

CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY
CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY
CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCC	000	000 PPP	PPP YYY	YYY
CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY
CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY
CCCCCCCCCCCC	0000000000	PPPPPPPPPPPPP	YYY	YYY

CCCCCCCC	000000	PPPPPPP PPPPPPP	YY YY	YY YY	SSSSSSS SSSSSSS	PPPPPPP PPPPPPP	EEEEEEE EEEEEEE	CCCCCCC CCCCCCC	SSSSSSS SSSSSSS	.....
CC	00	00	PP PP	PP YY	SS SS	PP PP	EE EE	CC	SS SS	....
CC	00	00	PP PP	PP YY	SS SS	PP PP	EE EE	CC	SS SS	....
CC	00	00	PP PP	YY YY	SS SS	PP PP	EE EE	CC	SS SS	....
CC	00	00	PPPPPPP PPPPPPP	YY YY	SSSSSS SSSSSS	PPPPPPP PPPPPPP	EEEEEEE EEEEEEE	CC	SSSSSS SSSSSS	....
CC	00	00	PPPPPPP PPPPPPP	YY YY	SSSSSS SSSSSS	PPPPPPP PPPPPPP	EEEEEEE EEEEEEE	CC	SSSSSS SSSSSS	....
CC	00	00	PP PP	YY YY	SS SS	PP PP	EE EE	CC	SS SS	....
CC	00	00	PP PP	YY YY	SS SS	PP PP	EE EE	CC	SS SS	....
CC	00	00	PP PP	YY YY	SS SS	PP PP	EE EE	CC	SS SS	....
CCCCCCCC	000000	PP PP	YY YY	SSSSSSS SSSSSSS	PP PP	EEEEEEE EEEEEEE	CCCCCCC CCCCCCC	SSSSSSS SSSSSSS	.....	....
CCCCCCCC	000000	PP PP	YY YY	SSSSSSS SSSSSSS	PP PP	EEEEEEE EEEEEEE	CCCCCCC CCCCCCC	SSSSSSS SSSSSSS	.....	....

LL		SSSSSSS SSSSSSS	.....
LL		SS SS	....
LLLLLLLL		SSSSSSS SSSSSSS	....
LLLLLLLL		SSSSSSS SSSSSSS	....

```
1 0001 0 MODULE copyspecs ( ! Manipulates input and output specifications for COPY utility
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 )
5 0005 1 BEGIN
6 0006 1
7 0007 1 ****
8 0008 1 ****
9 0009 1 ****
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****
30 0030 1 ****
31 0031 1 ++
32 0032 1 FACILITY: COPY Command
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module obtains input and output specifications from the CLI and opens
37 0037 1 the associated files.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 VAX/VMS operating system, unprivileged user mode utility,
42 0042 1 operates at non-AST level.
43 0043 1
44 0044 1 --
45 0045 1 ++
46 0046 1
47 0047 1 AUTHOR: Carol Peters. CREATION DATE: 14 April 1978 14:17
48 0048 1
49 0049 1 Modified by:
50 0050 1
51 0051 1 V03-011 TSK0010 Tamar Krichevsky 8-May-1984
52 0052 1 Rearrange the calls to CLISGET VALUE and LIBSFIND FILE, for
53 0053 1 input filename processing. This will fix the problem of
54 0054 1 COPY a.a.a.a.a.a NL: copying every other file, instead of
55 0055 1 every file.
56 0056 1
57 0057 1 V03-010 TSK0009 Tamar Krichevsky 20-Apr-1984
```

58 0058 1 | Before the input file is opened, clear the longest record  
59 0059 1 | length field in the input file's file header XAB. This will  
60 0060 1 | insure that the LRL value will be correct for record oriented  
61 0061 1 | devices. RMS does not clear this field if it is inappropriate.  
62 0062 1 | As a result, the LRL could be carried from one file to another.  
63 0063 1 | For example, given the command -- COPY foo.txt,SYSSINPUT a.a --  
64 0064 1 | SYSSINPUT inherited the LRL form foo.txt. (Not kosher!)

65 0065 1 |  
66 0066 1 | V03-009 TSK0008 Tamar Krichevsky 28-Mar-1984  
67 0067 1 | Fix IF statement in COPY\$OPN\_OUTFIL which sets up the  
68 0068 1 | default name string as ";" . It was broken by TSK007.  
69 0069 1 |  
70 0070 1 | V03-008 TSK0007 Tamar Krichevsky 2-Mar-1984  
71 0071 1 | Convert input file parsing and searching to LIBSFINDFILE.  
72 0072 1 | Place the check for WILD\_OUTPUT before the potential reparse  
73 0073 1 | of the output file. RMS changed how it set the bits in the  
74 0074 1 | NAMSL\_FNB field.  
75 0075 1 |  
76 0076 1 | V03-007 TSK0006 Tamar Krichevsky 16-Feb-1984  
77 0077 1 | Copy the input and output file names from the command line  
78 0078 1 | into the appropriate buffers. They were getting lost and  
79 0079 1 | some error messages were being displayed like so:  
80 0080 1 | "Error opening as input"  
81 0081 1 |  
82 0082 1 | Also add in check to see if the input file's record format  
83 0083 1 | is VFC and the fixed control region size is zero. The SOS  
84 0084 1 | editor created files like this. It knew that the smallest  
85 0085 1 | fixed header size was two bytes; so it assumed 2 when it  
86 0086 1 | saw 0. RMS compensated for this by setting the size to  
87 0087 1 | two bytes. Unfortunately, the incompatible attributes  
88 0088 1 | comparison would fail because the input file's HSZ field in  
89 0089 1 | the XABFHC was zero, but the output file's HSZ was two.  
90 0090 1 | When COPY encounters such an input file, it will change the  
91 0091 1 | HSZ field to two.  
92 0092 1 |  
93 0093 1 | V03-005 Tamar Krichevsky 3-Oct-1983  
94 0094 1 | Move the SDISPLAY, which was added in V03-005, to after the  
95 0095 1 | the check for a successful file SCREATE or \$OPEN. Otherwise,  
96 0096 1 | an extra message is issued when the file can not be accessed  
97 0097 1 | for the SDISPLAY.  
98 0098 1 |  
99 0099 1 | V03-005 LMPI0150 L. Mark Pilant, 9-Sep-1983 11:19  
100 0100 1 | Add a SDISPLAY to COPY\$OPN\_OUTFIL so that the protection  
101 0101 1 | of the created file may be obtained.  
102 0102 1 |  
103 0103 1 | V03-004 TSK0004 Tamar Krichevsky 8-Aug-1983  
104 0104 1 | Fix ACCVIO during append operations. Output file's XABPRO  
105 0105 1 | should not be removed from XAB chain until file is closed.  
106 0106 1 |  
107 0107 1 | V03-003 TSK0004 Tamar Krichevsky 8-Aug-1983  
108 0108 1 | Modify COPY\$OPN\_OUTFILE, SETUP OUTXAB and APPLY\_OUT QUAL so  
109 0109 1 | that file protection and revision information is not propagated  
110 0110 1 | to the output file from the input file. Fix bug which clears  
111 0111 1 | the expiration date when the output device is mag-tape. Fix  
112 0112 1 | bug in /PROTECTION qualifier so that unspecified fields are  
113 0113 1 | left alone.  
114 0114 1 |

115 0115 1 | V03-002 TSK0003 Tamar Krichevsky 4-Feb-1982  
116 0116 1 | Change over to the new CLI. Move external declarations from  
117 0117 1 | COPY.REQ into this module.  
118 0118 1 |  
119 0119 1 | V03-001 TSK0002 Tamar Krichevsky 4-Feb-1982  
120 0120 1 | Copy the buckets size from the input FAB in the output XAB to  
121 0121 1 | insure that the file is created with the correct bucket size.  
122 0122 1 | When a file is created, if there are any allocation XABs, the  
123 0123 1 | bucket size in the FAB is ignored. Therefore, if the input file  
124 0124 1 | has several areas, and area 0 does not have largest BKZ, something  
125 0125 1 | other than the BKZ in the first (and only, in COPY's case) XABALL  
126 0126 1 | must be used. The largest bucket size is kept in the input file's  
127 0127 1 | FAB. \*\*\*\*\* NOTE: This works only if the ISAM files (the  
128 0128 1 | worst offenders) are copied block mode. IF FOR ANY REASON ISAM FILES  
129 0129 1 | ARE COPIED USING RECORD MODE IN THE FUTURE, THIS PROCEDURE WILL HAVE TO  
130 0130 1 | BE CHANGED.  
131 0131 1 |  
132 0132 1 | X00025 TSK0001 Tamar Krichevsky 5-Feb-1982  
133 0133 1 | Have Global Buffer Count (GBC) transferred from input FAB to  
134 0134 1 | outout FAB.  
135 0135 1 |  
136 0136 1 | X00024 KRM0038 Karl Malik 12-Jan-1982  
137 0137 1 | Warn the user (in COPY\$OPN\_OUTFIL) if the output file  
138 0138 1 | was forced to stream format ( in a network copy to  
139 0139 1 | a 10,20 or RT system ).  
140 0140 1 |  
141 0141 1 | X00023 KRM0035 Karl Malik 31-Dec-1981  
142 0142 1 | Check for network quoted string in single output filespec  
143 0143 1 | & if found, do not force multiple output files.  
144 0144 1 |  
145 0145 1 | X00022 WMC0030 Wayne Cardoza 15-Dec-1981  
146 0146 1 | Disallow output directory wildcards remaining after the output  
147 0147 1 | file parse with the related input file.  
148 0148 1 |  
149 0149 1 | X00021 WMC0021 Wayne Cardoza 8-Dec-1981  
150 0150 1 | Set no\_output\_spec if only directory is wild and no explicit  
151 0151 1 | filename components.  
152 0152 1 |  
153 0153 1 | X00020 KFH0001 Ken Henderson 28-Sep-1981  
154 0154 1 | Expiration and Backup dates are not copied from input file,  
155 0155 1 | but instead are defaulted.  
156 0156 1 |  
157 0157 1 | X00019 WMC0001 Wayne Cardoza 22-Jul-1981  
158 0158 1 | Explicit protection specification should not cause old dates  
159 0159 1 | to be preserved if a file spec is also present.  
160 0160 1 |  
161 0161 1 | X00018 SPF0001 S. Forgey 27-Jan-1981  
162 0162 1 | Allow wildcard directories in output file specifications to  
163 0163 1 | go along with RMS now handling "sticky" directories.  
164 0164 1 |  
165 0165 1 | X00017 JAK0017 J. Krycka 18-Sep-1980  
166 0166 1 | Alter the X00006 special check for network access in setting up  
167 0167 1 | the output Allocation XAB (i.e., get ALQ and DEQ values from the  
168 0168 1 | FHC XAB).  
169 0169 1 |  
170 0170 1 | X00016 TMH0015 Tim Halvorsen 24-Mar-1980  
171 0171 1 | Force creation of a new file (creation date, owner, prot)

172 0172 1 | if the output file specification is explicit to maintain compatibility with release 1 behavior. This involves changing the previous update to remove remove xabpro.rdt.dat if explicit output filespec as long as /PROT was not specified (If /PROT specified, xabpro must not be removed to allow it to work).

173 0173 1 |

174 0174 1 |

175 0175 1 |

176 0176 1 |

177 0177 1 |

178 0178 1 |

179 0179 1 | X00015 TMH0014 Tim Halvorsen 19-Mar-1980  
180 0180 1 | Do not remove output XABPRO,RDT,DAT blocks if concat follows flag is set because we were only trying to prevent changing characteristics on existing files -- concatenation always produces a new file. Also, inhibit wildcard directories on output file specifications.

181 0181 1 |

182 0182 1 |

183 0183 1 |

184 0184 1 |

185 0185 1 |

186 0186 1 | X00014 TMH0013 Tim Halvorsen 17-Mar-1980  
187 0187 1 | Issue ENDPRM2 call at the same time as ENDPRM1 call to eliminate problems with parameter ordering (in MCR, the parameters appear in reverse order).

188 0188 1 |

189 0189 1 |

190 0190 1 |

191 0191 1 | X00013 JAK0003 J. Krycka 14-Jan-1980  
192 0192 1 | Undo X00005 change so that COPY will be able to use block I/O to copy relative and indexed files over the network.

193 0193 1 |

194 0194 1 |

195 0195 1 | X00012 TMH0012 T. Halvorsen 29-Dec-1979  
196 0196 1 | Remove XABPRO on appends since changing both owner or protection is prohibited (see X00010)

197 0197 1 |

198 0198 1 |

199 0199 1 | X00011 TMH0011 T. Halvorsen 15-Nov-1979  
200 0200 1 | Call CLI back with ENDPRM2 after output filespec is obtained to signal any unprocessed qualifiers.

201 0201 1 |

202 0202 1 |

203 0203 1 | X00010 TMH0010 T. Halvorsen 13-Nov-1979  
204 0204 1 | Zero the owner UIC field of the XABPRO on appends since changing the owner UIC for an existing file is prohibited.

205 0205 1 |

206 0206 1 |

207 0207 1 | X00009 TMH0009 T. Halvorsen 24-Oct-1979  
208 0208 1 | Test for output spec of only an explicit nodename so that the filename is defaulted correctly.  
209 0209 1 | Fix relative volume placement control to be hard (issue an error if the file cannot completely be placed on the volume).

210 0210 1 |

211 0211 1 |

212 0212 1 |

213 0213 1 | X00008 T. Halvorsen 25-Jul-1979  
214 0214 1 | Add relative volume placement control.  
215 0215 1 | Fix message to indicate contiguous-best-try is being tried when there is not enough contiguous space rather than issuing an error message.

216 0216 1 |

217 0217 1 |

218 0218 1 |

219 0219 1 | X00007 T. Halvorsen 14-Jul-1979  
220 0220 1 | Fix problem copying ISAM files after another file (BIO was left on from previous file).

221 0221 1 |

222 0222 1 |

223 0223 1 | X00006 JAK0002 J. Krycka 16-Mar-1978 14:00  
224 0224 1 | To support copy of files over the network, get ALQ and DEQ values from input XABALL if NET bit is set.

225 0225 1 |

226 0226 1 |

227 0227 1 | X00005 JAK0001 J. Krycka 16-Mar-1978 14:00  
228 0228 1 | To support copy of relative files over the network, set

229	0229	1		BRO bit in output FAB if NET bit is set.
230	0230	1		
231	0231	1		X00004 CHP20339 C. Peters 25-Oct-1978 14:10
232	0232	1		In COPY\$GET_INFIL\$ zero ESL and RSL fields to avoid
233	0233	1		reporting wrong file specification on error.
234	0234	1		
235	0235	1		X00003 CHP19547 C. Peters 7-Oct-1978 14:27
236	0236	1		Don't make version numbers sticky in an APPEND command.
237	0237	1		
238	0238	1	--	

```
: 240      0239 1 | Table of Contents
: 241      0240 1 |
: 242      0241 1 |
: 243      0242 1 FORWARD ROUTINE
: 244      0243 1   copy$get_infile,
: 245      0244 1   copy$opn_infile,
: 246      0245 1   copy$get_outfil,
: 247      0246 1   copy$opn_outfil,
: 248      0247 1   setup_exfend,
: 249      0248 1   setup_outxab
: 250      0249 1   apply_out_qual : NOVALUE;
: 251      0250 1   : NOVALUE;
: 252      0251 1 |
: 253      0252 1 | Include files
: 254      0253 1 |
: 255      0254 1 |
: 256      0255 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
: 257      0256 1 REQUIRE 'SRC$:COPYMSG.REQ';
: 258      0337 1 |
: 259      0338 1 |
: 260      0339 1 | Macros
: 261      0340 1 |
: 262      0341 1 MACRO
: 263      0342 1 |
: 264      0343 1   ! Check to see if the global or local qualifier flag is set without the
: 265      0344 1   ! local negation flag being set.
: 266      0345 1   !  
M 0346 1   qualifier_active( global_qual, local_qual, locally_negated ) =
: 268      M 0347 1   (IF (.global_qual AND NOT .locally_negated) OR .local_qual
: 269      M 0348 1   THEN true
: 270      0349 1   ELSE )%
: 271      0350 1   :
: 272      0351 1 |
: 273      0352 1 |
: 274      0353 1 | External variables
: 275      0354 1 |
: 276      0355 1 EXTERNAL
: 277      0356 1   copy$cli_status : $BBLOCK,
: 278      0357 1   copy$sem_status : $BBLOCK,
: 279      0358 1 |
: 280      0359 1   curr_allocation_value,
: 281      0360 1   curr_extension_value,
: 282      0361 1   curr_protection_or,
: 283      0362 1   curr_protection_and,
: 284      0363 1   curr_file_max_value,
: 285      0364 1   curr_volume_value,
: 286      0365 1 |
: 287      0366 1   infile_cli_desc : $BBLOCK[],
: 288      0367 1   in_name_desc : VECTOR,
: 289      0368 1   out_name_desc : VECTOR
: 290      0369 1   ;
: 291      0370 1 |
: 292      0371 1 REQUIRE
: 293      0372 1   'SRC$:COPY.REQ'
: 294      0373 1   :| Obtains the input file specification
| Opens the current input file
| Obtains the output file specification
| Opens the current output file
| Sets up an output file to be extended.
| Sets up XAB fields for an output file.
| Sets output fields depending on file qualifiers.
| VAX/VMS system definitions
| Definition of macros to SIGNAL a message
```

COPYSPECS  
V04-000

L 14  
15-Sep-1984 23:42:51  
15-Sep-1984 22:42:03

VAX-11 Bliss-32 V4.0-742  
\$255\$DUA28:[COPY.SRC]VMSMAC.REQ;1

Page 7  
(1)

: %PRINT: File: VMSMAC.B32, Version V04-000, Edit 1, WWC, 09-JAN-1978

: 295	0828	1	
: 296	0829	1	EXTERNAL ROUTINE
: 297	0830	1	cli\$get_value : addressing_model( general ),
: 298	0831	1	copy\$get_global_qual,
: 299	0832	1	copy\$get_local_qual,
: 300	0833	1	copy\$check_file_for_match,
: 301	0834	1	copy\$calc_qty,
: 302	0835	1	copy\$close_outf,
: 303	0836	1	copy\$inopn_err,
: 304	0837	1	copy\$log_msg,
: 305	0838	1	copy\$oclose_err,
: 306	0839	1	copy\$outopn_err,
: 307	0840	1	copy\$find_input_file,
: 308	0841	1	copy\$semantics;

Retrieves command level qualifiers  
Retrieves local qualifiers  
See if input file matches command line criteria  
Calculates a file extension quantity.  
Closes an output file  
Handles an input \$OPEN error  
Logs a message about COPY's activities  
Handles an output file close error.  
Handles an output \$OPEN error  
Finds and parses an input file specification  
Determines semantics of a command

```
310      0842 1 GLOBAL ROUTINE copy$get_infile (input_fab, input_nam, input_xaball) =  
311      0843 1           ! Obtain input file specification  
312      0844 1  
313      0845 1  
314      0846 1     ++ Functional description:  
315      0847 1  
316      0848 1     This routine gets an input file specification and all  
317      0849 1     related qualifiers from the Command Language Interpreter. Then  
318      0850 1     the file specification is parsed.  
319      0851 1  
320      0852 1     If a wildcard specification is still being processed, or if  
321      0853 1     no more input specifications are available, this routine just  
322      0854 1     returns successfully.  
323      0855 1  
324      0856 1     A series of flags are set if certain conditions obtain. These  
325      0857 1     conditions describe the current list of files that are candidates  
326      0858 1     for concatenation. The flags are set if the file specification  
327      0859 1     contains input wildcards, an explicit wildcard version number, or an explicit version number.  
328      0860 1  
329      0861 1     Another flag applies only to this specification and says whether it contains any wildcards.  
330      0862 1  
331      0863 1     Calling sequence:  
332      0864 1         copy$get_infile (input_fab.ra.v, input_nam.ra.v, input_xaball.ra.v)  
333      0865 1  
334      0866 1  
335      0867 1     Input parameters:  
336      0868 1  
337      0869 1         input_fab      - the FAB to use for this input specification  
338      0870 1         input_nam      - the NAM to use for this input specification  
339      0871 1         input_xaball   - the XABALL to use for this input specification  
340      0872 1  
341      0873 1     Implicit inputs:  
342      0874 1  
343      0875 1         wildcard_active - a bit in COPY$CLI_STATUS that says that we are  
344      0876 1         already processing an input wildcard.  
345      0877 1  
346      0878 1     Output parameters:  
347      0879 1         none  
348      0880 1  
349      0881 1  
350      0882 1  
351      0883 1     Implicit outputs:  
352      0884 1         The fields of the FAB and the NAM block are filled in according  
353      0885 1         to the CLI call and the SPARSE function call.  
354      0886 1  
355      0887 1         The RSL field of the dummy nam blk is filled in by the routine COPY$FIND_INPUT_FILE. This is later  
356      0888 1         used in parsing the name additional input files or output files.  
357      0889 1  
358      0890 1     A bit in COPY$CLI_STATUS may be set:  
359      0891 1  
360      0892 1         multiple_input - more than one input file specification in the command  
361      0893 1         wildcard_active - if a wildcard is present  
362      0894 1  
363      0895 1  
364      0896 1  
365      0897 1     Some bits in COPY$SEM_STATUS may be set:  
366      0898 1         wild_input    - wildcard fields exist  
                         wild_inp_ver - a wildcard version number exists
```

```
367      0899 1 | exp_inp_ver - an explicit version number exists
368      0900 1 |
369      0901 1 | Routine value:
370      0902 1 |
371      0903 1 |   OK          - success
372      0904 1 |   NO_MORE_FILES - success, no more input specifications
373      0905 1 |   NO_FILE     - failure
374      0906 1 |
375      0907 1 | Side effects:
376      0908 1 |
377      0909 1 |   none
378      0910 1 |
379      0911 1 | --
380      0912 1 |
381      0913 2 | BEGIN
382      0914 2 |
383      0915 2 | LOCAL
384      0916 2 |   rtn_status;           ! Retrun status from external calls
385      0917 2 |
386      0918 2 | MAP
387      0919 2 |   input_fab    : REF BLOCK [, BYTE],       ! FAB to use with input file
388      0920 2 |   input_nam    : REF BLOCK [, BYTE],       ! NAM to use with input file
389      0921 2 |   input_xaball : REF BLOCK [, BYTE];     ! XABALL to use with input file
390      0922 2 |
391      0923 2 |
392      0924 2 |
393      0925 2 | Return if a wildcard file specification is currently being processed or the
394      0926 2 | last input file name has been retrieved from the command line. Otherwise,
395      0927 2 | set the flag which indicates that more input files have been found.
396      0928 2 |
397      0929 2 |
398      0930 2 | IF .wildcard_active
399      0931 2 | THEN
400      0932 2 |   RETURN ok;           ! If a wildcard specification is currently
401      0933 2 | being processed, then just return to caller.
402      0934 2 |
403      0935 2 | Reinitialize the RSL and ESL fields of the NAM block so that a parsing
404      0936 2 | error does not report an error in the previous file processed.
405      0937 2 |
406      0938 2 |
407      0939 2 | input_nam [nam$b_esl] = 0;           ! Expanded string length of zero.
408      0940 2 | input_nam [nam$b_rsl] = 0;           ! Resultant string length of zero.
409      0941 2 |
410      0942 2 |
411      0943 2 |
412      0944 2 | Call LIB$FIND_FILE to parse the input file specification. This resolves
413      0945 2 | logical names and determines if there are wildcards present, or explicit
414      0946 2 | named fields present.
415      0947 2 |
416      0948 2 |
417      0949 2 | IF NOT (rtn_status = copy$find_input_file ( infile_cli_desc ))
418      0950 2 | THEN
419      0951 2 |   IF .rtn_status NEQ RMSS_NMF
420      0952 2 |   THEN
421      0953 2 |     RETURN .rtn_status;
422      0954 2 |
423      0955 2 | !
```

```
424      0956 2    ! Initialize the input file FAB.  
425      0957 2  
426      0958 2  
427      P 0959 2 SFAB_INIT (          ! Setup the input file FAB as follows:  
428      P 0960 2   FAB = .input_fab,  ! FAB address is the input parameter  
429      P 0961 2   FAC = <GET,BRO>, ! Input file, mixed block and record acce  
430      P 0962 2   SHR = GET,     ! Allow others to read the input file  
431      P 0963 2   DNA = 0,      ! No default file specification  
432      P 0964 2   RTV = 0,      ! Use default retrieval window size  
433      P 0965 2   RAT = CR,     ! Carriage control in case unit record input  
434      P 0966 2   FOP = <S00,NAM>, ! Sequential I/O only, open by name block  
435      P 0967 2   NAM = .input_nam, ! NAM block address  
436      P 0968 2   XAB = .input_xaball); ! XABALL block address.  
437      0969 2  
438      0970 2  
439      0971 2    ! If there were no more files for the current inout specification, get the next  
440      0972 2    one from the command line.  
441      0973 2  
442      0974 2    IF .rtn_status EQ RMSS_NMF  
443      0975 2    THEN  
444      0976 2    BEGIN  
445      0977 2  
446      0978 2    IF NOT (rtn_status = CLISGET_VALUE( SDESCRIPTOR('infile'), infile_cli_desc))  
447      0979 2    THEN  
448      0980 2    RETURN no_more_files;  
449      0981 2  
450      0982 2    ! Get the qualifiers for this input file.  
451      0983 2  
452      0984 2    COPY$GET_LOCAL_QUAL();  
453      0985 2  
454      0986 2    ! Check to see if more than one input file has been given.  
455      0987 2  
456      0988 2    IF .rtn_status NEQ SSS_NORMAL  
457      0989 2    THEN  
458      0990 2    multiple_input = TRUE;  
459      0991 2  
460      0992 2  
461      0993 2    ! Reinitialize the RSL and ESL fields of the NAM block so that a parsing  
462      0994 2    error does not report an error in the previous file processed.  
463      0995 2  
464      0996 2  
465      0997 2    input_nam [nam$b_est] = 0;          ! Expanded string length of zero.  
466      0998 2    input_nam [nam$b_rsl] = 0;        ! Resultant string length of zero.  
467      0999 2  
468      1000 2  
469      1001 2  
470      1002 2    ! Call LIB$FIND_FILE to parse the input file specification. This resolves  
471      1003 2    logical names and determines if there are wildcards present, or explicit  
472      1004 2    named fields present.  
473      1005 2  
474      1006 2  
475      1007 2    IF NOT (rtn_status = copy$find_input_file ( infile_cli_desc ))  
476      1008 2    THEN  
477      1009 2    RETURN .rtn_status;  
478      1010 2    END;  
479      1011 2  
480      1012 2    !
```

```

481      1013 2 | Now test the type of expanded name string that we have. Does it contain wildcards? Were
482      1014 2 | certain fields explicitly named?
483      1015 2 |
484      1016 2 |
485      1017 2 | IF .input_nam [nam$v_wildcard]
486      1018 2 | THEN
487      1019 2 |   BEGIN
488      1020 2 |     wildcard_active = TRUE;
489      1021 2 |     wild_input = TRUE;
490      1022 2 |     first_wild_infile = TRUE;
491      1023 2 |   END
492      1024 2 |
493      1025 2 |
494      1026 2 | ELSE
495      1027 2 |   wildcard_active = FALSE;
496      1028 2 |
497      1029 2 | IF .input_nam [nam$v_wild_ver]
498      1030 2 | THEN
499      1031 2 |   wild_inp_ver = TRUE
500      1032 2 |
501      1033 2 |
502      1034 2 | ELSE
503      1035 2 |   BEGIN
504      1036 2 |     IF .input_nam [nam$v_exp_ver]
505      1037 2 |     THEN
506      1038 2 |       exp_inp_ver = TRUE;
507      1039 2 |
508      1040 2 |
509      1041 2 |
510      1042 2 |
511      1043 2 | RETURN ok;
512      1044 1 | END;

| If there were any wildcards,
| set WILDCARD_ACTIVE. This says current file
| specification contains wildcards.
| Also set WILD_INPUT. This says that the current
| input list contains wildcard specs somewhere.
| Indicate this is the first wild input file
| If no input wildcards in this spec, turn off
| the WILDCARD_ACTIVE flag.
| If an explicit wildcard version number
| was specified,
| set the WILD_INP_VER flag.
| Otherwise,
| see if an explicit version number was specified
| If it is, set the EXP_INP_VER flag, meaning
| that there is an explicit input version number.

| Return with success.

|
```

<pre> 65 6C 69 66 6E 69 00000 P.AAB: 00006                                 00000006 00008 P.AAA: 00000000 0000C </pre>	<pre> .TITLE COPYSPEC .IDENT \V04-000\  .PSECT SPLITS,NOWRT,NOEXE,2  .ASCII \infile\ .BLKB 2 .LONG 6 .ADDRESS P.AAB  .EXTRN COPY\$MSG_NUMBER .EXTRN COPY\$CLI_STATUS .EXTRN COPY\$SEM_STATUS .EXTRN CURR_ALLOCATION_VALUE .EXTRN CURR_EXTENSION_VALUE .EXTRN CURR_PROTECTION_OR .EXTRN CURR_PROTECTION_AND .EXTRN CURR_FILE_MAX_VALUE .EXTRN CURR_VOLUME_VALUE .EXTRN INFICE CLI DESC .EXTRN IN_NAME_DESC, OUT_NAME_DESC .EXTRN CLIS_PRESENT, CLIS_NEGATED .EXTRN CLIS_LOCRES, CLIS_LOCNEG </pre>
--	---

E 15  
15-Sep-1984 23:42:51 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:14:19 [COPY.SRC][COPYSPECs.B32;1]

		09	35	A7	E9	000A7	6\$:	BLBC	53(R7), 7\$	: 1017
		69	02200010	8F	C8	000AB		BISL2	#35651600, COPY\$SEM_STATUS	: 1024
		05	02	04	11	000B2		BRB	8\$	: 1017
			34	20	8A	000B4	7\$:	BICB2	#32, COPY\$SEM_STATUS+2	: 1027
				03	F1	000B8	8\$:	BBC	#32, 52(R7), 9\$	: 1029
				69	20	88	000BD	BISB2	#32, COPY\$SEM_STATUS	: 1031
				03	07	11	000C0	BRB	10\$	: 1034
				69	A7	E9	000C2	BLBC	52(R7), 10\$	: 1036
				50	02	88	000C6	BISB2	#2, COPY\$SEM_STATUS	: 1043
					01	D0	000C9	MOVL	#1, R0	: 1044
					04	000CC	10\$:	RET		

; Routine Size: 205 bytes, Routine Base: SCODES + 0000

```
1045 1 GLOBAL ROUTINE copy$opn_infile (input_fab) = ! Open the current input file
1046 1
1047 1 //+
1048 1 Functional description:
1049 1
1050 1 This routine opens the current input file. If the input file
1051 1 specification contains a wildcard field, an RMS $SEARCH for the
1052 1 next wildcard match occurs before the actual file open.
1053 1
1054 1 Any input parameter qualifiers are applied to the file's RMS blocks before
1055 1 the open is performed. For now, the only valid qualifier is /READ_CHECK.
1056 1
1057 1 If the OPEN fails, an error is reported to SYS$ERROR. When input wildcards are present,
1058 1 two types of failure are permitted:
1059 1
1060 1 RMSS_NMF - no more files match given wildcard
1061 1 open failure - allowed when a file matching a wildcard spec cannot be
1062 1 opened, as long as that file would have been copied without concatenation.
1063 1
1064 1 Calling sequence:
1065 1
1066 1     copy$opn_infile (input_fab.ra.v)
1067 1
1068 1 Input parameters:
1069 1
1070 1     input_fab - the FAB associated with the input file
1071 1
1072 1 Implicit inputs:
1073 1
1074 1     COPY$CLI_STATUS bits are checked:
1075 1
1076 1         read_check_bit - This bit is set if the /READ_CHECK qualifier was specified for this file.
1077 1         wildcard_active - This specification contains wildcards.
1078 1             Find the next file with a $SEARCH function call.
1079 1
1080 1         input file NAM block is read to obtain the length of the resultant name string
1081 1         input file XABFH to check the HSZ for VFC files.
1082 1
1083 1     COPY$SEM_STATUS bits are checked:
1084 1
1085 1         multiple_output - Multiple files are being produced. This is checked to allow for
1086 1             open failure on a wildcard specified file.
1087 1
1088 1 Output parameters:
1089 1
1090 1     none
1091 1
1092 1 Implicit outputs:
1093 1
1094 1     in_name_desc - the length field of the input name descriptor is written from the RSL
1095 1         field in the NAM block
1096 1
1097 1     The FABSV_RCK bit in the input FAB is set if /READ_CHECK was specified.
1098 1
1099 1     COPY$CLI_STATUS bit settings may be altered:
1100 1
1101 1     wildcard_active - turned off if no more files that match wildcard are found.
```

```
571      1102 1 |     infile_open    - set if the file is opened successfully
572      1103 1 |
573      1104 1 | Routine value:
574      1105 1 |
575      1106 1 |     OK          - input file open
576      1107 1 |     NO_MORE_FILES - no further wildcard match found
577      1108 1 |     NO_WILD_OPEN   - open failure on wildcard match file
578      1109 1 |     NO_FILE       - input file not found
579      1110 1 |
580      1111 1 | Side effects:
581      1112 1 |
582      1113 1 | The input file is opened.
583      1114 1 | If an RMS SEARCH function fails, then an error is reported on SYS$ERROR.
584      1115 1 |
585      1116 1 |--|
586      1117 1 | BEGIN
587      1118 2 |
588      1119 2 | MAP
589      1120 2 |     input_fab      : REF BLOCK [, BYTE];           ! input FAB block
590      1121 2 |
591      1122 2 | BIND
592      1123 2 |     input_xaball   =
593      1124 2 |     .input_fab [fab$1_xab] : BLOCK [, BYTE];   ! input file XABALL block
594      1125 2 |
595      1126 2 |     input_xabdat   =
596      1127 2 |     .input_xaball [xab$1_nxt] : BLOCK [, BYTE]; ! input file XABDAT block
597      1128 2 |     input_xabfhc   =
598      1129 2 |     .input_xabdat [xab$1_nxt] : BLOCK [, BYTE]; ! input file XABFHC block
599      1130 2 |     input_nam      =
600      1131 2 |     .input_fab [fab$1_nam] : BLOCK [, BYTE];   ! input NAM block address
601      1132 2 |
602      1133 2 | LOCAL
603      1134 2 |     status;                                ! RMS status code variable
604      1135 2 |
605      1136 2 |
606      1137 2 | If a wildcard specification is active, call RMS to search for the next wildcard match.
607      1138 2 |
608      1139 2 |
609      1140 2 | IF .wildcard_active                         ! If an input wildcard field is present,
610      1141 2 | THEN
611      1142 2 |     IF NOT .first_wild_infile
612      1143 2 |     THEN
613      1144 2 |         BEGIN
614      1145 2 |             status = COPY$FIND_INPUT_FILE( infile_cli_desc );
615      1146 2 |
616      1147 2 |             IF .status EQL rms$_nmf           ! If no more wildcard matches exist,
617      1148 2 |             THEN
618      1149 2 |                 BEGIN
619      1150 2 |                     wildcard_active = FALSE;
620      1151 2 |                     RETURN no_more_files;
621      1152 2 |                     END;
622      1153 2 |
623      1154 2 |             IF NOT .status
624      1155 2 |             THEN
625      1156 2 |                 BEGIN
626      1157 2 |                     copy$inopn_err (.input_fab);
627      1158 2 |                     wildcard_active = FALSE;
628      1159 2 |                 END;
629      1160 2 |             END;
630      1161 2 |         END;
631      1162 2 |     END;
632      1163 2 | 
```

```
628      1159      RETURN no_file;
629      1160      END;
630      1161      END;
631      1162      ELSE
632      1163      first_wild_infile = FALSE;
633      1164
634      1165
635      1166      ! If the user specified the input read checking qualifier, turn on the appropriate bit in the FAB.
636      1167
637      1168
638      1169      IF qualifier_active( read_chk_qual, loc_read_chk_qual, neg_read_chk_qual)
639      1170      THEN
640      1171      input_fab [fab$V_rck] = TRUE
641      1172      ! then turn on the FAB read check indicator.
642      1173      ELSE
643      1174      input_fab [fab$V_rck] = FALSE;
644      1175      ! Otherwise, turn it off.
645      1176      ! Open the input file. First, zero the LRL field in the file header XAB. This
646      1177      insures that it will have the appropriate value if the input device is record
647      1178      oriented (i.e. SYSSINPUT).
648      1179
649      1180
650      1181      input_xabfhc[ XABSW_LRL ] = 0;
651      P 1182      IF $RMS_OPEN (
652      P 1183      FAB = .input_fab,
653      1184      ERR = copy$inopn_err)
654      1185      THEN
655      1186      BEGIN
656      1187      infile_open = TRUE;
657      1188      in_name_desc [0] = .input_nam [nam$B_rsl];
658      1189      ! indicate that the file is open
659      1190      ! and set the length of the input file name descr
660      1191      ! If record format is VFC and the HSZ is 0, then set the HSZ to 2.
661      1192      ! If this isn't done, the incompatible attributes check will
662      1193      ! incorrectly fail.
663      1194      IF .input_fab [FABSB_RFM] EQL FABSC_VFC
664      1195      AND
665      1196      .input_xabfhc [XABSB_HSZ] EQL 0
666      1197      THEN
667      1198      input_xabfhc [XABSB_HSZ] = 2;
668      1199      RETURN ok;
669      1200      END
670      1201
671      1202      ELSE
672      1203      BEGIN
673      1204
674      1205
675      1206      ! If multiple output files are being produced, and this is a file that matches a wildcard specification,
676      1207      ! allow the open to fail. This means that one file that matches the wildcard specification is not copied
677      1208      ! to a new output file.
678      1209
679      1210
680      1211      IF .wildcard_active AND
681      1212      (.multiple_output OR NOT .explicit_concat_qual )
682      1213      THEN
683      1214      RETURN no_wild_open
684      1215
```

```
685      1216 3      RETURN no_file;
686      1217 2      END;
687      1218 2      END;
688      1219 1      END;
```

					.EXTRN SYSSOPEN	
					.ENTRY COPYSOPN_INFILE. Save R2,R3,R4,R5,R6	1045
					COPYSCLI_STATUS+4, R6	
					COPYSSEM_STATUS, R5	
					INPUT_FAB R2	
					36(R2), R6	1125
					4(R0). R0	
					4(R0). R3	1127
					40(R2), R4	1129
					#5. COPYSSEM_STATUS+2. 38	1131
					#1. COPYSSEM_STATUS+3, 28	1140
					INFILE CLI DESC	1142
					#1. COPY\$FIND INPUT_FILE	1145
					STATUS, #99018	
					1S	1147
					#32. COPYSSEM_STATUS+2	1150
					#3. R0	1151
					RET	
					BLBS STATUS. 38	1154
					R2	1157
					PUSHL	
					CALLS #1. COPYSINOPN_ERR	
					BICB2 #32. COPYSSEM_STATUS+2	1158
					BRB	1159
					11\$	
					BICB2 #2. COPYSSEM_STATUS+3	1163
					COPYSCLI_STATUS+4, 48	1169
					BBC #2. COPY\$CLI_STATUS+4, 58	
					BBC #1. COPY\$CLI_STATUS+4, 68	
					BISB2 #128. 6(R2)	1171
					BRB 7S	
					BICB2 #128. 6(R2)	1173
					CLRW 10(R3)	1181
					PUSHAB COPYSINOPN_ERR	1184
					R2	
					PUSHL #2. SYSSOPEN	
					CALLS R0. 98	
					BLBC	
					BISB2 #4. COPYSSEM_STATUS+2	1187
					MOVZBL 3(R4) IN_NAME_DESC	1188
					CMP8 31(R2), #3	1194
					BNEQ 8S	
					TSTB 23(R3)	1196
					BNEQ 8S	
					MOVBL #2. 23(R3)	1198
					RET #1. R0	1203
					BBC #5. COPYSSEM_STATUS+2 118	
					BLBS COPYSSEM_STATUS+1, 108	1211
					BBS #2. COPY\$CLI_STATUS. 118	1212
					MOVL #5. R0	1216
					RET	

COPYSPECS  
V04-000

K 15  
15-Sep-1984 23:42:51  
14-Sep-1984 12:14:19

VAX-11 Bliss-32 V4.0-742  
[COPY.SRC]COPYSPECS.B32;1

Page 19  
(4)

50 D4 000B1 11\$: CLRL R0  
04 000B3 RET

; 1219

; Routine Size: 180 bytes, Routine Base: SCODES + 00CD

```
1220 1 GLOBAL ROUTINE copy$get_outfil (output_fab, output_nam, output_xabfhc) =  
1221 1 ! Obtain the output file specification  
1222 1  
1223 1 ++ Functional description:  
1224 1  
1225 1 This routine obtains the output file specification and all  
1226 1 related qualifiers from the Command Language Interpreter. Then  
1227 1 the file specification is parsed without any help from related input file name  
1228 1 blocks. This initial parse determines whether the file specification had null file  
1229 1 name, type, and version number fields.  
1230 1  
1231 1 If no output file name, type, or version number is given, a flag  
1232 1 is set in COPY$SEM_STATUS.  
1233 1  
1234 1 Calling sequence:  
1235 1  
1236 1 copy$get_outfil (output_fab.ra.v, output_nam.ra.v, output_xabfhc.ra.v)  
1237 1  
1238 1 Input parameters:  
1239 1  
1240 1 output_fab - the FAB to use for this output specification  
1241 1 output_nam - the NAM to use for this output specification  
1242 1 output_xabfhc - the XABFHC to use for this output specification  
1243 1  
1244 1 Implicit inputs:  
1245 1  
1246 1 The RLF field of the output NAM block contains the address of the input file NAM block.  
1247 1  
1248 1  
1249 1 Output parameters:  
1250 1  
1251 1 1252 1 none  
1253 1  
1254 1 Implicit outputs:  
1255 1  
1256 1 The fields of the FAB and the NAM block are filled in according  
1257 1 to the CLI call, FAB initialization, and the SPARSE function call.  
1258 1  
1259 1 A bit may be set in COPY$SEM_STATUS:  
1260 1  
1261 1 no_output_spec - no output name, type, or version number specified.  
1262 1  
1263 1 Routine value:  
1264 1  
1265 1 OK - success  
1266 1 NO_FILE - the SPARSE function call returned an error code  
1267 1  
1268 1 Side effects:  
1269 1  
1270 1 An error is reported if the SPARSE function returns an error status code and  
1271 1 COPY$OUTOPN_ERR is called.  
1272 1 --  
1273 1  
1274 2 BEGIN  
1275 2 MAP  
1276 2
```

```

: 747      1277 2      output_fab    : REF BLOCK [, BYTE],          ! FAB to use with output file
: 748      1278 2      output_nam   : REF BLOCK [, BYTE],          ! NAM to use with output file
: 749      1279 2      output_xabfhc : REF BLOCK [, BYTE];        ! XABFHc to use with output file
: 750      1280 2
: 751      1281 2
: LOCAL
: 752      1282 2      cli_desc : $BBLOCK[ DSC$C_S_BLN ].           ! Descriptor for qualifier values
: 753      1283 2      temp_rlf;                                ! Holds the output RLF field
: 754
: 755
: 756
: 757      1287 2      ! Initialize descriptor. Retrieve the output file specification.
: 758      1288 2
: 759      1289 2      CHSFILL( 0, DSC$C_S_BLN, cli_desc );
: 760      1290 2      cli_desc[ DSC$B_CLASS ] = DSC$K_CLASS_D;
: 761      1291 2
: 762      1292 2      CLISGET_VALUE( SDESCRIPTOR('OUTFILE'), cli_desc );
: 763      1293 2
: 764      1294 2      ! Save the file name in the output name descriptor; in case the name
: 765      1295 2      doesn't parse. The name given on the command line will be used
: 766      1296 2      in the error message returned to the user.
: 767      1297 2
: 768      1298 2      out_name_desc[0] = .cli_desc[DSC$W_LENGTH];
: 769      1299 2      CHSMOVE(.cli_desc[DSC$W_LENGTH], .cli_desc[DSC$A_POINTER], .out_name_desc[1]);
: 770      1300 2
: 771      1301 2      ! Get the qualifiers for the output file.
: 772      1302 2
: 773      1303 2      COPY$GET_GLOBAL_QUAL();
: 774      1304 2
: 775      1305 2
: 776      1306 2      ! Initialize the output file FAB.
: 777      1307 2
: 778      1308 2
: P 1309 2      SFAB_INIT (
: P 1310 2      FAB = .output_fab,
: P 1311 2      FAC = <PUT,TRN>,
: P 1312 2      SHR = NIL,
: P 1313 2      FNA = .cli_desc[DSC$A_POINTER],
: P 1314 2      FNS = .cli_desc[DSC$W_LENGTH],
: P 1315 2      RTV = 0,
: P 1316 2      FOP = <SOO,OPF,NAM>,
: P 1317 2      NAM = .output_nam,
: P 1318 2      XAB = .output_xabfhc);                         ! Setup the output file FAB as follows:
: 779
: 780
: 781
: 782
: 783
: 784
: 785
: 786
: 787
: 788
: 789
: 790
: 791
: 792
: 793
: 794
: 795
: 796
: 797
: 798
: 799
: 800
: 801
: 802
: 803      1319 2      ! FAB address is the output parameter
: 1320 2      Output file
: 1321 2      No file sharing
: 1322 2      File name address from CLI
: 1323 2      File name size from CLI also
: 1324 2      Use the system default retrieval window size
: 1325 2      Sequential operations only, output file parse,
: 1326 2      NAM block address
: 1327 2      XABFHc block address
: 1328 2      name block open
: 1329 2
: 1330 2
: 1331 2
: 1332 2
: 1333 2      Zero the expanded string length so that the COPY error routine, copy$outopn_err, can
: 1322 2      decide if an expanded name string was created by RMS.
: 1323 2
: 1324 2
: 1325 2
: 1326 2      output_nam[nam$b_esl] = 0;                            ! Zero the output expanded string length.
: 1327 2
: 1328 2
: 1329 2
: 1330 2
: 1331 2
: 1332 2
: 1333 2      Temporarily remove the RLF field of the output NAM block so that the
: 1322 2      output file specification can be tested for null name, type, and
: 1323 2      version number fields.
: 1324 2
: 1325 2
: 1326 2
: 1327 2
: 1328 2
: 1329 2
: 1330 2
: 1331 2
: 1332 2
: 1333 2

```

```
804      1334 2      temp_rlf = .output_nam [nam$1_rlf];           ! Save the RLF field because it may be needed later.  
805      1335 2      output_nam [nam$1_rlf] = 0;                 ! Set the RLF field to null.  
806  
807  
808  
809  
810  
811      P 1341  
812      P 1342  
813      1343  
814      1344  
815      1345  
816      1346  
817      1347  
818      1348  
819      1349  
820      1350  
821      1351  
822      1352  
823      1353  
824      1354  
825      1355  
826      1356  
827      1357  
828      1358  
829      1359  
830      1360  
831      1361  
832      1362  
833      1363  
834      1364  
835      1365  
836      1366  
837      1367  
838      1368  
839      1369  
840      1370  
841      1371  
842      1372  
843      1373  
844      1374  
845      1375  
846      1376  
847      1377  
848      1378  
849      1379  
850      1380  
851      1381  
852      1382  
853      1383  
854      1384  
855      1385  
856      1386  
857      1387  
858      1388  
859      1389  
860      1390  
          | Parse the output file specification.  
          |  
          | IF NOT $RMS_PARSE ( FAB = .output_fab,  
          |                      ERR = copy$outopen_err)  
          | THEN  
          |     RETURN no_file;  
          |  
          | Test for an absence of the file name, type, and version number fields  
          | (or the presence of a network quoted string).  
          |  
          | IF (NOT .output_nam [nam$1_wild_name]) AND  
          |     (NOT .output_nam [nam$1_wild_type]) AND  
          |     (NOT .output_nam [nam$1_wild_ver]) AND  
          |     (NOT .output_nam [nam$1_quoted]) AND  
          |     (NOT .output_nam [nam$1_exp_name]) AND  
          |     (NOT .output_nam [nam$1_exp_type]) AND  
          |     (NOT .output_nam [nam$1_exp_ver]) AND  
          |     (.output_nam [nam$1_exp_dir] OR  
          |      .output_nam [nam$1_exp_dev] OR  
          |      .output_nam [nam$1_node])  
          | THEN  
          |     no_output_spec = TRUE;  
          |  
          | If no output wildcards are present,  
          | and no quoted string  
          | and no output name,  
          | and no output type,  
          | and no output version number,  
          | and an explicit directory  
          | or device name  
          | or node name is given,  
          | then set NO_OUTPUT_SPEC bit.  
          |  
          | If the file name, file type or version fields are ALL either wild or no specified and  
          | the output file spec does not contain a quoted string, then set the flag which indicates  
          | that the output file spec was completely wild.  
          |  
          | IF (.output_nam [nam$1_wild_name] OR NOT .output_nam [nam$1_exp_name])  
          |     AND  
          |     (.output_nam [nam$1_wild_type] OR NOT .output_nam [nam$1_exp_type])  
          |     AND  
          |     (.output_nam [nam$1_exp_ver] OR NOT .output_nam [nam$1_wild_ver])  
          |     AND  
          |     NOT .output_nam [nam$1_quoted]  
          | THEN  
          |     no_expl_out_fields = TRUE;  
          |  
          | Reload the RLF field. Another PARSE will be performed later in the routine  
          | COPY$OPEN_OUTFILE and may take fields from the input resultant file string.  
          |  
          | output_nam [nam$1_rlf] = .temp_rlf;  
          |  
          | Return with a success code.  
          |  
          | RETURN ok;  
          |  
          | ! Return successfully.
```

```
: 861      1391 2
: 862      1392 1      END;
```

										.PSECT SPLIT\$,NOWRT,NOEXE,2	
										P.AAD: .ASCII \OUTFILE\	
										.BLKB 1	
										.LONG 7	
										.ADDRESS P.AAD	
										.EXTRN SYSPARSE	
										.PSECT SCODE\$,NOWRT,2	
										.ENTRY COPY\$GET_OUTFIL, Save R2,R3,R4,R5,R6	1220
										SUBL2 #8, SP	
										MOVCS #0, (SP), #0, #8, CLI_DESC	1289
										MOVB #2, CLI_DESC+3	
										PUSHL SP	
										PUSHAB P.AAC	
										CALLS #2, CLISGET VALUE	
										CLI_DESC, 00T NAME DESC	
										CLI_DESC, @CLI_DESC+4, @OUT_NAME_DESC+4	1298
										MOVZUL CLI_DESC+4, @OUT_NAME_DESC+4	1299
										MOVCS #0, COPY\$GET GLOBAL_QUAL	1303
										OUTPUT FAB, R6	1318
										#0, (SP), #0, #80, (R6)	
08	00	SE	6E		007C 00000	08 C2 00002					
					00 2C 00005	00 2C 0000A					
					6E 0000A	02 90 0000B					
					5E DD 0000F	5E 9F 00011					
					C0 00015	02 FB 00015					
					6E 0001C	6E 3C 0001C					
					6E 00021	6E 28 00021					
					00 FB 00028	00 FB 00028					
					AC DO 0002D	AC DO 0002D					
					00 2C 00031	00 2C 00031					
					66 00038	66 00038					
					8F B0 00039	8F B0 00039					
					D0 0003E	MOVL #553648192, 4(R6)					
					B0 00046	MOVW #8209, 22(R6)					
					02 90 0004C	MOVB #2, 31(R6)					
					AC DO 00050	MOVL OUTPUT_XABFH, 36(R6)					
					DO 00055	MOVL OUTPUT_NAM, R2					
					52 DO 00059	MOVL R2, 40(R6)					
					AE DO 0005D	MOVL CLI_DESC+4, 44(R6)					
					6E 90 00062	MOVL CLI_DESC, 52(R6)					
					A2 94 00066	CLRB 11(R2)					
					D0 00069	MOVL 16(R2), TEMP_RLF					
					A2 D4 0006D	CLRL 16(R2)					
					CF 9F 00070	PUSHAB COPY\$OUTOPN_ERR					
					56 DD 00074	PUSHL R6					
					02 FB 00076	CALLS #2, SYSPARSE					
					50 E9 0007D	BLBC R0, 78					
					A2 9E 00080	MOVAB 52(R2), R0					
					50 E0 00084	BBS #5, (R0), 38					
					04 E0 00088	BBS #4, (R0), 28					
					03 E0 0008C	BBS #3, (R0), 28					
					12 E0 00090	BBS #2, (R0), 28					
					02 E0 00094	BBS #1, (R0), 28					
					01 E0 00098	BBS #0, (R0), 28					
					60 E8 0009C	BLBS (R0), 28					
					06 E0 0009F	BBS #6, (R0), 18					
					60 95 000A3	TSTB (R0)					
					04 19 000A5	BLSS 18					

05		60	11	E1 000A7	BBC	#17.	1361
	0000G	CF	08	88 000AB	1\$: BISB2	#8. CC \$SEM_STATUS	1363
04		60	05	E0 000B0	2\$: BBS	#5. (F1). 3\$	1370
18		60	02	E0 000B4	BBS	#2. (F2). 6\$	
04		60	04	E0 000B8	3\$: BBS	#4. (R2). 4\$	1372
10		60	01	E0 000BC	BBS	#1. (RU). 6\$	
		04	60	E8 000C0	4\$: BLBS	(R0) 5\$	1374
09		60	03	E0 000C3	BBS	#3. (RC). 6\$	
05		60	12	E0 000C7	5\$: BISB2	#18. (RCY). 6\$	1376
	0000G	CF	01	88 000CB	MOV L	#1. COPY\$SEM_STATUS+3	1378
		A2	53	D0 000D0	6\$: MOVL TEMP RLF, 16(R2)		1384
		50	01	D0 000D4	MOVL #1, R0		1390
			04	000D7	RET		
			50	D4 000D8	7\$: CLRL R0		
			04	000DA	RET		1392

; Routine Size: 219 bytes, Routine Base: SCODE\$ + 0181

864 1393 1 GLOBAL ROUTINE copy\$opn\_outfil (output\_fab, output\_rab, input\_fab, out\_file\_count) =  
865 1394 1 ! Opens the current output file  
866 1395 1  
867 1396 1 ++  
868 1397 1 Functional description:  
869 1398 1  
870 1399 1 This routine opens the current output file. If it is already open due  
871 1400 1 to input file concatenation, the output file RAB is simply disconnected from  
872 1401 1 the FAB to permit switching from block mode I/O to record mode I/O.  
873 1402 1  
874 1403 1 Many of the fields in the input FAB and XAB blocks are copied into the corresponding  
875 1404 1 output FAB and XAB blocks. Also, bits and values are set in the output XAB and FAB blocks  
876 1405 1 because of output file qualifiers specified on the command.  
877 1406 1  
878 1407 1 If the output file already exists, and is being overwritten, it is opened  
879 1408 1 for output. If the output file does not exist, it is allocated and then opened.  
880 1409 1  
881 1410 1 Calling sequence:  
882 1411 1  
883 1412 1 copy\$opn\_outfil (output\_fab.ra.v, output\_rab.ra.v, input\_fab.ra.v, out\_file\_count.wl.r)  
884 1413 1  
885 1414 1 Input parameters:  
886 1415 1  
887 1416 1 output\_fab - the address of the FAB associated with the output file  
888 1417 1 output\_rab - the address of the RAB to be used with the output file  
889 1418 1 input\_fab - the address of the FAB associated with the input file  
890 1419 1  
891 1420 1 Implicit inputs:  
892 1421 1  
893 1422 1 copy\$cli\_status - the OUTFILE\_OPEN bit indicates whether an output file is already open.  
894 1423 1 - bits indicate the settings of the output file qualifiers  
895 1424 1  
896 1425 1 Fields from the input NAM and XAB block are used in the output NAM and XAB blocks.  
897 1426 1  
898 1427 1  
899 1428 1 Output parameters:  
900 1429 1 out\_file\_count - a counter that is incremented if a new file is opened.  
901 1430 1  
902 1431 1 Implicit outputs:  
903 1432 1  
904 1433 1 copy\$cli\_status - OUTFILE\_OPEN is set once the file is opened.  
905 1434 1 - EXTEND\_OUTFILE is set if the output file is being extended.  
906 1435 1  
907 1436 1 Fields are written in the output\_fab and its associated NAM and XAB blocks.  
908 1437 1  
909 1438 1 out\_name\_desc - a descriptor for the output file. Its length field is written.  
910 1439 1  
911 1440 1 When the output file name is parsed, various bits are set in  
912 1441 1 COPYSEM\_STATUS. These include:  
913 1442 1  
914 1443 1 wild\_output - output spec includes explicit wildcards  
915 1444 1 wild\_out\_ver - explicit wildcard version number  
916 1445 1  
917 1446 1 Routine value  
918 1447 1  
919 1448 1 OK - output file successfully created or readied for more output  
920 1449 1 NO\_FILE - output file could not be opened, created, or readied for output

```

921      1450 1 | Side effects:
922      1451 1 |
923      1452 1 |
924      1453 1 | The routine SETUP_EXTEND is called if the output file is open. The value of this call
925      1454 1 | is returned to the caller.
926      1455 1 | The routine SETUP_OUTXAB is called to write most of the output XAB block fields.
927      1456 1 | Messages are output if a file was created during an APPEND command, if versions were
928      1457 1 | slipped under higher existing versions, or if files were replaced or overlaid.
929      1458 1 |
930      1459 1 | --
931      1460 1 |
932      1461 2 | BEGIN
933      1462 2 |
934      1463 2 | MAP
935      1464 2 |   output_fab : REF BLOCK [. BYTE].
936      1465 2 |   output_rab : REF BLOCK [. BYTE].
937      1466 2 |   input_fab : REF BLOCK [. BYTE].
938      1467 2 |   out_file_count : REF VECTOR;
939      1468 2 |           ! FAB to use with output file
940      1469 2 |           ! RAB to use with output file
941      1470 2 |           ! FAB of the current input file
942      1471 2 |           ! pointer to number of output files written
943      1472 2 | BIND
944      1473 2 |   output_nam =
945      1474 2 |       .output_fab [fab$1_nam]
946      1475 2 |   output_xabfch =
947      1476 2 |       .output_fab [fab$1_xab]
948      1477 2 |   output_xaball =
949      1478 2 |       .output_xabfch [xab$1_nxt]
950      1479 2 |   output_xabdat =
951      1480 2 |       .output_xaball [xab$1_nxt]
952      1481 2 |   output_xabrdt =
953      1482 2 |       .output_xabdat [xab$1_nxt]
954      1483 2 |           ! output NAM block address
955      1484 2 |           ! output XAB file header characteristics block
956      1485 2 |           ! output XAB date block
957      1486 2 |           ! output XAB date block
958      1487 2 |           ! output XAB date block
959      1488 2 |           ! output XAB date block
960      1489 2 |           ! output XAB date block
961      1490 2 |           ! output XAB date block
962      1491 2 | LOCAL
963      1492 2 |   status;           ! Status variable for calling semantic routine.
964      1493 2 | IF .outfile_open
965      1494 2 | THEN
966      1495 2 |   RETURN setup_extend (
967      1496 2 |           .output_rab);           ! If the output file is already open,
968      1497 2 |           ! call a routine to set the file up
969      1498 2 |           ! to be extended.
970      1499 2 | Copy a set of FAB attributes from the input to the output FAB.
971      1500 2 |   output_fab [fab$2_org] = .input_fab [fab$2_org];           ! The fields copied are file organization,
972      1501 2 |   output_fab [fab$2_rat] = .input_fab [fab$2_rat];           ! record attributes
973      1502 2 |   output_fab [fab$2_mrs] = .input_fab [fab$2_mrs];           ! maximum record size
974      1503 2 |   output_fab [fab$2_mrn] = .input_fab [fab$2_mrn];           ! maximum record number
975      1504 2 |   output_fab [fab$2_rf] = .input_fab [fab$2_rf];           ! record format
976      1505 2 |   output_fab [fab$2_fsz] = .input_fab [fab$2_fsz];           ! fixed control area size
977      1506 2 |   output_fab [fab$2_bks] = .input_fab [fab$2_bks];           ! bucket size

```

```
978      1507 2   output_fab [fab$w_gbc] = .input_fab [fab$w_gbc]; ! global buffer count
979      1508 2
980      1509 2
981      1510 2   | If the input file has read or write checking options, copy them to the output file.
982      1511 2
983      1512 2
984      1513 2   output_fab [fab$1_fop] = .output_fab [fab$1_fop] OR ! OR together the current FOP output field
985      1514 2   (.input_fab [fab$1_fop] AND (fab$1_rck OR fab$1_wck));
986      1515 2   ! and the read and write check bits of the
987      1516 2   ! FOP input field.
988      1517 2
989      1518 2
990      1519 2   | Decide on block or record I/O.
991      1520 2
992      1521 2
993      1522 2   IF .input_fab [fab$b_org] EQL fab$c_seq           ! If the input file is a sequential file,
994      THEN
995      1523 2   output_fab [fab$v_bro] = TRUE               ! then indicate mixed block and record I/O.
996      ELSE
997      1524 2   BEGIN
998      1525 2   output_fab [fab$v_bio] = true;
999      1526 2   output_fab [fab$v_bro] = false;             ! Otherwise, indicate only block I/O.
1000     1527 2
1001     1528 2
1002     1529 2
1003     1530 2
1004     1531 2
1005     1532 2   | Copy input blocksize for tapes. Otherwise let RMS set the output blocksize.
1006     1533 2
1007     1534 2
1008     1535 2   IF .input_fab [$FAB_DEV (sqd)]           ! If input device is a tape,
1009     THEN
1010     1536 2   output_fab [fab$w_bls] = .input_fab [fab$w_bls] ! then copy the blocksize to the output FAB.
1011     1537 2
1012     1538 2   ELSE
1013     1539 2   output_fab [fab$w_bls] = 0;                ! Otherwise, let RMS choose blocksize.
1014     1540 2
1015     1541 2
1016     1542 2   | Test the expanded name string for the output file. Does it contain wildcards? If so,
1017     1543 2   is there an explicit wildcard version number?
1018     1544 2
1019     1545 2
1020     1546 2   IF .output_nam [nam$v_wildcard]          ! If there were any wildcards,
1021     THEN
1022     1547 2   wild_output = TRUE;                  ! set flag saying that the file specification
1023     1548 2
1024     1549 2
1025     1550 2
1026     1551 2   IF .output_nam [nam$v_wild_ver]        ! If the version number is a wildcard,
1027     THEN
1028     1552 2   wild_out_ver = TRUE                 ! output version number, remember it.
1029     1553 2
1030     1554 2   ELSE
1031     1555 2   IF .output_nam [nam$v_exp_ver]       ! Otherwise, see if an explicit version number was s
1032     THEN
1033     1556 2   exp_out_ver = TRUE;                  ! If so, set the EXP_OUT_VER flag.
1034     1557 2
1035     1558 2
1036     1559 2
1037     1560 2
1038     1561 2   | Reparse the output string with a wildcard version number, if this is not
1039     1562 2   an APPEND operation and one of the following cases is true:
1040     - no output file name, type or version number was given
1041     (e.g. [COPY x.x [dir]])
```

```
1035      1564 2 | - wild or explicit version numbers were given for the input file, but
1036      1565 2 |   the version field for the output file was not specified
1037      1566 2 |   (e.g. COPY x.x;* a.a)
1038      1567 2 |   - the output spec is wild (e.g. COPY x.x *, or COPY x.x *.*)
1039      1568 2 |
1040      1569 2 | IF NOT .append command
1041          AND
1042          (.no_output_spec
1043              OR
1044                  (.wild_inp_ver OR .exp_inp_ver)
1045                  AND NOT .output_nam [nam$v_wild_ver]
1046                  AND NOT .output_nam [nam$v_exp_ver])
1047                      OR
1048                          (NOT .output_nam [nam$v_exp_ver]
1049                          AND (.output_nam [nam$v_wild_type] OR NOT .output_nam [nam$v_exp_type])
1050                          AND .output_nam [nam$v_wild_name]))
1051      1570 3 | THEN
1052          BEGIN
1053              output_fab [fab$1_dna] = UPLIT (';+');
1054              output_fab [fab$2_dns] = 2;           ! Then provide a default name string
1055          END;                                ! of an explicit output wildcard
1056
1057
1058      1577 4 | ! Now SPARSE (this may be a reparse) the output file specification.
1059
1060      1578 4 | IF NOT $RMS_PARSE ( FAB = .output_fab, ERR = copy$outopn_err)
1061      1579 3 | THEN
1062          RETURN no_file;                     ! On failure, return with an error code.
1063
1064
1065      1580 3 | ! No director wildcards allowed to remain at this time
1066      1581 3 |
1067      1582 3 |
1068      1583 3 |
1069      1584 3 |
1070      1585 3 |
1071      1586 3 |
1072      1587 3 |
1073      1588 3 |
1074      1589 3 |
1075      1590 3 |
1076      1591 3 |
1077      1592 3 |
1078      1593 3 |
1079      1594 3 |
1080      1595 3 |
1081      1596 3 |
1082      1597 3 |
1083      1598 3 |
1084      1599 3 |
1085      1600 3 |
1086      1601 3 |
1087      1602 3 |
1088      1603 4 |
1089      1604 4 |
1090      1605 4 |
1091      1606 4 |
1092      1607 4 |
1093      P 1608 4 | LOCAL
1094          outputstr : vector[2];
1095          outputstr[0] = .output_nam [nam$2_esl];
1096          outputstr[1] = .output_nam [nam$1_esl];
1097          PUT_MESSAGE( MSGS_SYNTAX,
1098              1,
1099              outputstr,
1100              0 );
1101          RETURN no_file;
1102      END;
1103
1104      1610 4 |
1105      1611 4 |
1106      1612 4 |
1107      1613 3 |
1108      1614 3 |
1109      1615 2 |
1110      1616 2 |
1111      1617 2 |
1112      1618 2 |
1113      1619 2 |
1114      1620 2 | See if the output file fits the criteria given on the command line.
1115
1116      IF NOT (status = copy$check_file_for_match())
1117      THEN
```

```

1092      1621 2      RETURN .status;
1093      1622 2
1094      1623 2
1095      1624 2      | Call the routine SETUP_OUTXAB to copy output XAB fields from the corresponding input XAB fields.
1096      1625 2
1097      1626 2
1098      1627 2      setup_outxab (
1099      1628 2          .output_fab,
1100      1629 2          .input_fab);
1101      1630 2
1102      1631 2
1103      1632 2      | Write output XAB fields by calling
1104      1633 2          a routine that selects the necessary fields fro
1105      1634 2          the input FAB and writes them into the output F
1106      1635 2
1107      1636 2      | Call the routine APPLY_OUT_QUAL to write RMS fields according to output parameter qualifiers.
1108      1637 2
1109      1638 2
1110      1639 2      apply_out_qual (
1111      1640 2          .output_fab);
1112      1641 2
1113      1642 2
1114      1643 2      | Process output file qualifiers
1115      1644 2
1116      1645 2
1117      1646 2
1118      1647 2      | Call the routine COPY$SEMANTICS to determine the semantic effects of
1119      1648 2          this particular combination of input and output file specifications and qualifiers.
1120      1649 2
1121      1650 2
1122      1651 2      IF NOT copy$semantics (
1123      1652 2          copy$sem_status,
1124      1653 2          .input_fab,
1125      1654 3          .output_fab)
1126      1655 3
1127      1656 3
1128      1657 3      THEN
1129      1658 2          | Decide what semantic behavior is required.
1130      1659 2          Pass the status variable copy$sem_status,
1131      1660 2          the input FAB block address,
1132      1661 3          and the output FAB block address.
1133      1662 3          | If the input/output spec combination makes no sens
1134      1663 3          then return with error status code.
1135      1664 2
1136      1665 2
1137      1666 2
1138      1667 2      | Perform special XAB setup if a concatenated file is being created.
1139      1668 2
1140      1669 2
1141      1670 2      IF (.append_command
1142      1671 2          OR .concat_follows
1143      1672 2          OR NOT .no_expl_out_fields
1144      1673 2          OR NOT .input_fab [$fab_dev (fod)])
1145      1674 2          THEN
1146      1675 2          output_xaball [xab$l_nxt] = .output_xabrdt [xab$l_nxt] ! Do not provide any date information
1147      1676 2          ELSE
1148      1677 2              BEGIN
1149      1678 3                  output_xaball [xab$l_nxt] = output_xabdat;
1150      1679 3                  output_xabdat [xab$l_nxt] = output_xabrdt;
1151      1680 2              END;
1152      1681 2
1153      1682 2      | If appending to existing file,
1154      1683 2          or concatenating
1155      1684 2          or if explicit field in output spec
1156      1685 2          or the input device is not file structured,
1157      1686 2
1158      1687 2
1159      1688 2      | Create (or simply open) the output file.
1160      1689 2
1161      1690 2      extend_outfile = FALSE;
1162      1691 2
1163      1692 2
1164      1693 2      | Assume that the output file is not being extended.
1165      1694 2
1166      1695 2
1167      1696 2
1168      1697 2      | If a file needn't be created, just open an existing file.
1169      1698 2
1170      1699 2
1171      1700 2
1172      1701 2
1173      1702 2
1174      1703 2
1175      1704 2
1176      1705 2      IF .append_command AND
1177      1706 2          NOT .new_version_qual
1178      1707 2
1179      1708 2
1180      1709 2
1181      1710 2
1182      1711 2
1183      1712 2
1184      1713 2
1185      1714 2
1186      1715 2
1187      1716 2
1188      1717 2
1189      1718 2
1190      1719 2
1191      1720 2
1192      1721 2
1193      1722 2
1194      1723 2
1195      1724 2
1196      1725 2
1197      1726 2
1198      1727 2
1199      1728 2
1200      1729 2
1201      1730 2
1202      1731 2
1203      1732 2
1204      1733 2
1205      1734 2
1206      1735 2
1207      1736 2
1208      1737 2
1209      1738 2
1210      1739 2
1211      1740 2
1212      1741 2
1213      1742 2
1214      1743 2
1215      1744 2
1216      1745 2
1217      1746 2
1218      1747 2
1219      1748 2
1220      1749 2
1221      1750 2
1222      1751 2
1223      1752 2
1224      1753 2
1225      1754 2
1226      1755 2
1227      1756 2
1228      1757 2
1229      1758 2
1230      1759 2
1231      1760 2
1232      1761 2
1233      1762 2
1234      1763 2
1235      1764 2
1236      1765 2
1237      1766 2
1238      1767 2
1239      1768 2
1240      1769 2
1241      1770 2
1242      1771 2
1243      1772 2
1244      1773 2
1245      1774 2
1246      1775 2
1247      1776 2
1248      1777 2
1249      1778 2
1250      1779 2
1251      1780 2
1252      1781 2
1253      1782 2
1254      1783 2
1255      1784 2
1256      1785 2
1257      1786 2
1258      1787 2
1259      1788 2
1260      1789 2
1261      1790 2
1262      1791 2
1263      1792 2
1264      1793 2
1265      1794 2
1266      1795 2
1267      1796 2
1268      1797 2
1269      1798 2
1270      1799 2
1271      1800 2
1272      1801 2
1273      1802 2
1274      1803 2
1275      1804 2
1276      1805 2
1277      1806 2
1278      1807 2
1279      1808 2
1280      1809 2
1281      1810 2
1282      1811 2
1283      1812 2
1284      1813 2
1285      1814 2
1286      1815 2
1287      1816 2
1288      1817 2
1289      1818 2
1290      1819 2
1291      1820 2
1292      1821 2
1293      1822 2
1294      1823 2
1295      1824 2
1296      1825 2
1297      1826 2
1298      1827 2
1299      1828 2
1300      1829 2
1301      1830 2
1302      1831 2
1303      1832 2
1304      1833 2
1305      1834 2
1306      1835 2
1307      1836 2
1308      1837 2
1309      1838 2
1310      1839 2
1311      1840 2
1312      1841 2
1313      1842 2
1314      1843 2
1315      1844 2
1316      1845 2
1317      1846 2
1318      1847 2
1319      1848 2
1320      1849 2
1321      1850 2
1322      1851 2
1323      1852 2
1324      1853 2
1325      1854 2
1326      1855 2
1327      1856 2
1328      1857 2
1329      1858 2
1330      1859 2
1331      1860 2
1332      1861 2
1333      1862 2
1334      1863 2
1335      1864 2
1336      1865 2
1337      1866 2
1338      1867 2
1339      1868 2
1340      1869 2
1341      1870 2
1342      1871 2
1343      1872 2
1344      1873 2
1345      1874 2
1346      1875 2
1347      1876 2
1348      1877 2
1349      1878 2
1350      1879 2
1351      1880 2
1352      1881 2
1353      1882 2
1354      1883 2
1355      1884 2
1356      1885 2
1357      1886 2
1358      1887 2
1359      1888 2
1360      1889 2
1361      1890 2
1362      1891 2
1363      1892 2
1364      1893 2
1365      1894 2
1366      1895 2
1367      1896 2
1368      1897 2
1369      1898 2
1370      1899 2
1371      1900 2
1372      1901 2
1373      1902 2
1374      1903 2
1375      1904 2
1376      1905 2
1377      1906 2
1378      1907 2
1379      1908 2
1380      1909 2
1381      1910 2
1382      1911 2
1383      1912 2
1384      1913 2
1385      1914 2
1386      1915 2
1387      1916 2
1388      1917 2
1389      1918 2
1390      1919 2
1391      1920 2
1392      1921 2
1393      1922 2
1394      1923 2
1395      1924 2
1396      1925 2
1397      1926 2
1398      1927 2
1399      1928 2
1400      1929 2
1401      1930 2
1402      1931 2
1403      1932 2
1404      1933 2
1405      1934 2
1406      1935 2
1407      1936 2
1408      1937 2
1409      1938 2
1410      1939 2
1411      1940 2
1412      1941 2
1413      1942 2
1414      1943 2
1415      1944 2
1416      1945 2
1417      1946 2
1418      1947 2
1419      1948 2
1420      1949 2
1421      1950 2
1422      1951 2
1423      1952 2
1424      1953 2
1425      1954 2
1426      1955 2
1427      1956 2
1428      1957 2
1429      1958 2
1430      1959 2
1431      1960 2
1432      1961 2
1433      1962 2
1434      1963 2
1435      1964 2
1436      1965 2
1437      1966 2
1438      1967 2
1439      1968 2
1440      1969 2
1441      1970 2
1442      1971 2
1443      1972 2
1444      1973 2
1445      1974 2
1446      1975 2
1447      1976 2
1448      1977 2
1449      1978 2
1450      1979 2
1451      1980 2
1452      1981 2
1453      1982 2
1454      1983 2
1455      1984 2
1456      1985 2
1457      1986 2
1458      1987 2
1459      1988 2
1460      1989 2
1461      1990 2
1462      1991 2
1463      1992 2
1464      1993 2
1465      1994 2
1466      1995 2
1467      1996 2
1468      1997 2
1469      1998 2
1470      1999 2
1471      2000 2
1472      2001 2
1473      2002 2
1474      2003 2
1475      2004 2
1476      2005 2
1477      2006 2
1478      2007 2
1479      2008 2
1480      2009 2
1481      2010 2
1482      2011 2
1483      2012 2
1484      2013 2
1485      2014 2
1486      2015 2
1487      2016 2
1488      2017 2
1489      2018 2
1490      2019 2
1491      2020 2
1492      2021 2
1493      2022 2
1494      2023 2
1495      2024 2
1496      2025 2
1497      2026 2
1498      2027 2
1499      2028 2
1500      2029 2
1501      2030 2
1502      2031 2
1503      2032 2
1504      2033 2
1505      2034 2
1506      2035 2
1507      2036 2
1508      2037 2
1509      2038 2
1510      2039 2
1511      2040 2
1512      2041 2
1513      2042 2
1514      2043 2
1515      2044 2
1516      2045 2
1517      2046 2
1518      2047 2
1519      2048 2
1520      2049 2
1521      2050 2
1522      2051 2
1523      2052 2
1524      2053 2
1525      2054 2
1526      2055 2
1527      2056 2
1528      2057 2
1529      2058 2
1530      2059 2
1531      2060 2
1532      2061 2
1533      2062 2
1534      2063 2
1535      2064 2
1536      2065 2
1537      2066 2
1538      2067 2
1539      2068 2
1540      2069 2
1541      2070 2
1542      2071 2
1543      2072 2
1544      2073 2
1545      2074 2
1546      2075 2
1547      2076 2
1548      2077 2
1549      2078 2
1550      2079 2
1551      2080 2
1552      2081 2
1553      2082 2
1554      2083 2
1555      2084 2
1556      2085 2
1557      2086 2
1558      2087 2
1559      2088 2
1560      2089 2
1561      2090 2
1562      2091 2
1563      2092 2
1564      2093 2
1565      2094 2
1566      2095 2
1567      2096 2
1568      2097 2
1569      2098 2
1570      2099 2
1571      2100 2
1572      2101 2
1573      2102 2
1574      2103 2
1575      2104 2
1576      2105 2
1577      2106 2
1578      2107 2
1579      2108 2
1580      2109 2
1581      2110 2
1582      2111 2
1583      2112 2
1584      2113 2
1585      2114 2
1586      2115 2
1587      2116 2
1588      2117 2
1589      2118 2
1590      2119 2
1591      2120 2
1592      2121 2
1593      2122 2
1594      2123 2
1595      2124 2
1596      2125 2
1597      2126 2
1598      2127 2
1599      2128 2
1600      2129 2
1601      2130 2
1602      2131 2
1603      2132 2
1604      2133 2
1605      2134 2
1606      2135 2
1607      2136 2
1608      2137 2
1609      2138 2
1610      2139 2
1611      2140 2
1612      2141 2
1613      2142 2
1614      2143 2
1615      2144 2
1616      2145 2
1617      2146 2
1618      2147 2
1619      2148 2
1620      2149 2
1621      2150 2
1622      2151 2
1623      2152 2
1624      2153 2
1625      2154 2
1626      2155 2
1627      2156 2
1628      2157 2
1629      2158 2
1630      2159 2
1631      2160 2
1632      2161 2
1633      2162 2
1634      2163 2
1635      21
```

I 16  
15-Sep-1984 23:42:51 VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 12:14:19 [COPY.SRC]COPYSPEC\$.\$32;1

```

1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1678 2
1679 3
1680 4
1681 4
1682 4
1683 4
1684 4
1685 4
1686 4
1687 4
1688 4
1689 4
1690 4
1691 4
1692 4
1693 4
1694 4
1695 4
1696 4
1697 4
1698 4
1699 4
1700 4
1701 4
1702 4
1703 4
1704 4
1705 4
1706 3
1707 3
1708 3
1709 3
1710 3
1711 3
1712 3
1713 3
1714 3
1715 3
1716 3
1717 3
1718 3
1719 3
1720 3
1721 3
1722 3
1723 3
1724 3
1725 3
1726 3
1727 4
1728 4
1729 4
1730 4
1731 4
1732 4
1733 4
1734 4
THEN
    BEGIN
        IF NOT (status = SRMS_OPEN ( FAB = .output_fab, ERR = copySoutopn_err))
        THEN RETURN .status;
    END
ELSE
    BEGIN
        status = SRMS_CREATE (FAB = .output_fab);           ! Else, create (or open if it exists) file
    END

    If the file could not be created as a contiguous file because the disk was too full,
    then try to create it contiguous best try.

    IF .status EQ rms$_ful
        AND .output_xaball [xab$v_ctg]
        AND NOT qualifier_active("contig_qual", loc_config_qual, neg_config_qual )
    THEN
        BEGIN
            output_xaball [xab$v_ctg] = FALSE;           ! then turn off the contiguous indicator,
            output_xaball [xab$v_cbt] = TRUE;             ! turn on the contiguous best try indicator,
            status = SRMS_CREATE (                           ! and retry the create.
                FAB = .output_fab,                         ! Specify the address of the FAB block
                ERR = copySoutopn_err);                   ! if config-best-try ok.

            IF .status
                THEN put_message (msg$_cbt);           ! then issue message
            END                                         ! and an error action routine.

            ELSE
                IF NOT .status
                    THEN copySoutopn_err (.output_fab); ! Else, if error,
                                                ! issue error message
                END
        END

    Change the RMS return status to "created" if indeed the file was created.

    IF NOT .output_fab [fab$v_cif] AND
        .status EQ rms$_normal
    THEN
        status = rms$_created;                      ! Since RMS returns RMSS_NORMAL whether or not the
                                                    ! file was created, for internal reporting, change
                                                    ! the status code to RMSS_CREATED if appropriate.
                                                    ! (I.e., if the file was created.)

    If the file was indeed created, issue a SDISPLAY to obtain information
    about the newly created file.

    IF NOT .status
    THEN
        RETURN no_file;                            ! If the open or create failed,
                                                    ! then return an error status code.

    IF NOT SRMS_DISPLAY (FAB = .output_fab)
    THEN
        copySoutopn_err (.output_fab);

    If the output file was copied to a 10,20 or RT node and it was forced to a
    stream format file, then (if the /LOG qualifier was specified) warn the user
    of the conversion.

```

```

1206      1735 3   IF .status EQL rms$_cre_stm AND .LOG_MSG_QUAL
1207      1736 3   THEN
1208      1737 4   BEGIN
1209      1738 4   out_name_desc [0] = .output_nam [nam$b_rsl]; ! Store the length of the filespec
1210      P 1739 4   put_message (msg$_createdstm,1, ! Issue the message
1211          1740 4           out_name_desc);
1212          1741 4   status = rms$_created; ! Change the status as above
1213          1742 3   END;
1214          1743 2   END;
1215          1744 2
1216          1745 2
1217          1746 2   outfile_open = TRUE; ! Otherwise, set a flag saying that an output file is
1218          1747 2   out_name_desc [0] = .output_nam [nam$b_rsl]; ! and store the length of the file specification.
1219          1748 2
1220          1749 2   ! Clean up the output open procedure by reporting to the user if necessary and
1221          1750 2   updating more fields.
1222          1751 2
1223          1752 2
1224          1753 2   SELECTONE .status OF
1225          1754 2
1226          1755 2   SET
1227          1756 2
1228          1757 2   [rms$ created]:
1229          1758 2   BEGIN
1230          1759 3   out_file_count [0] =
1231          1760 3       .out_file_count [0] + 1; ! Update count of files created.
1232          1761 3
1233          1762 3   IF .append_command
1234          1763 3   THEN
1235          1764 3   copy$log_msg (
1236          1765 3           msg$_created); ! If this is an APPEND command,
1237          1766 3
1238          1767 3   IF .output_nam [nam$v_highver] AND ! send the following message to the user:
1239          1768 3       NOT .quiet_slip ! "<file-name> created" because creation is u
1240          1769 3   THEN
1241          P 1770 3   put_message (
1242          P 1771 3           msg$_highver, 1, ! send the following message to the user:
1243          1772 3           out_name_desc); ! ">higher version of <file-name> exists"
1244          1773 3
1245          1774 2   END;
1246          1775 2
1247          1776 2
1248          1777 2   [rms$ supersede]:
1249          1778 2   BEGIN
1250          1779 3   out_file_count [0] =
1251          1780 3       .out_file_count [0] + 1; ! Update count of files created.
1252          1781 3
1253          1782 3   copy$log_msg (
1254          1783 3           msg$_replaced); ! Send the following message to the user:
1255          1784 3
1256          1785 2
1257          1786 2
1258          1787 2
1259          1788 2
1260          1789 2
1261          1790 2   [rms$ normal]:
1262          1791 2   BEGIN
1263          1792 2   IF .append_command ! Output file existed previously and was opened.
1264          1793 2
1265          1794 2
1266          1795 2
1267          1796 2
1268          1797 2
1269          1798 2
1270          1799 2
1271          1800 2
1272          1801 2
1273          1802 2
1274          1803 2
1275          1804 2
1276          1805 2
1277          1806 2
1278          1807 2
1279          1808 2
1280          1809 2
1281          1810 2
1282          1811 2
1283          1812 2
1284          1813 2
1285          1814 2
1286          1815 2
1287          1816 2
1288          1817 2
1289          1818 2
1290          1819 2
1291          1820 2
1292          1821 2
1293          1822 2
1294          1823 2
1295          1824 2
1296          1825 2
1297          1826 2
1298          1827 2
1299          1828 2
1300          1829 2
1301          1830 2
1302          1831 2
1303          1832 2
1304          1833 2
1305          1834 2
1306          1835 2
1307          1836 2
1308          1837 2
1309          1838 2
1310          1839 2
1311          1840 2
1312          1841 2
1313          1842 2
1314          1843 2
1315          1844 2
1316          1845 2
1317          1846 2
1318          1847 2
1319          1848 2
1320          1849 2
1321          1850 2
1322          1851 2
1323          1852 2
1324          1853 2
1325          1854 2
1326          1855 2
1327          1856 2
1328          1857 2
1329          1858 2
1330          1859 2
1331          1860 2
1332          1861 2
1333          1862 2
1334          1863 2
1335          1864 2
1336          1865 2
1337          1866 2
1338          1867 2
1339          1868 2
1340          1869 2
1341          1870 2
1342          1871 2
1343          1872 2
1344          1873 2
1345          1874 2
1346          1875 2
1347          1876 2
1348          1877 2
1349          1878 2
1350          1879 2
1351          1880 2
1352          1881 2
1353          1882 2
1354          1883 2
1355          1884 2
1356          1885 2
1357          1886 2
1358          1887 2
1359          1888 2
1360          1889 2
1361          1890 2
1362          1891 2
1363          1892 2
1364          1893 2
1365          1894 2
1366          1895 2
1367          1896 2
1368          1897 2
1369          1898 2
1370          1899 2
1371          1900 2
1372          1901 2
1373          1902 2
1374          1903 2
1375          1904 2
1376          1905 2
1377          1906 2
1378          1907 2
1379          1908 2
1380          1909 2
1381          1910 2
1382          1911 2
1383          1912 2
1384          1913 2
1385          1914 2
1386          1915 2
1387          1916 2
1388          1917 2
1389          1918 2
1390          1919 2
1391          1920 2
1392          1921 2
1393          1922 2
1394          1923 2
1395          1924 2
1396          1925 2
1397          1926 2
1398          1927 2
1399          1928 2
1400          1929 2
1401          1930 2
1402          1931 2
1403          1932 2
1404          1933 2
1405          1934 2
1406          1935 2
1407          1936 2
1408          1937 2
1409          1938 2
1410          1939 2
1411          1940 2
1412          1941 2
1413          1942 2
1414          1943 2
1415          1944 2
1416          1945 2
1417          1946 2
1418          1947 2
1419          1948 2
1420          1949 2
1421          1950 2
1422          1951 2
1423          1952 2
1424          1953 2
1425          1954 2
1426          1955 2
1427          1956 2
1428          1957 2
1429          1958 2
1430          1959 2
1431          1960 2
1432          1961 2
1433          1962 2
1434          1963 2
1435          1964 2
1436          1965 2
1437          1966 2
1438          1967 2
1439          1968 2
1440          1969 2
1441          1970 2
1442          1971 2
1443          1972 2
1444          1973 2
1445          1974 2
1446          1975 2
1447          1976 2
1448          1977 2
1449          1978 2
1450          1979 2
1451          1980 2
1452          1981 2
1453          1982 2
1454          1983 2
1455          1984 2
1456          1985 2
1457          1986 2
1458          1987 2
1459          1988 2
1460          1989 2
1461          1990 2
1462          1991 2
1463          1992 2
1464          1993 2
1465          1994 2
1466          1995 2
1467          1996 2
1468          1997 2
1469          1998 2
1470          1999 2
1471          2000 2
1472          2001 2
1473          2002 2
1474          2003 2
1475          2004 2
1476          2005 2
1477          2006 2
1478          2007 2
1479          2008 2
1480          2009 2
1481          2010 2
1482          2011 2
1483          2012 2
1484          2013 2
1485          2014 2
1486          2015 2
1487          2016 2
1488          2017 2
1489          2018 2
1490          2019 2
1491          2020 2
1492          2021 2
1493          2022 2
1494          2023 2
1495          2024 2
1496          2025 2
1497          2026 2
1498          2027 2
1499          2028 2
1500          2029 2
1501          2030 2
1502          2031 2
1503          2032 2
1504          2033 2
1505          2034 2
1506          2035 2
1507          2036 2
1508          2037 2
1509          2038 2
1510          2039 2
1511          2040 2
1512          2041 2
1513          2042 2
1514          2043 2
1515          2044 2
1516          2045 2
1517          2046 2
1518          2047 2
1519          2048 2
1520          2049 2
1521          2050 2
1522          2051 2
1523          2052 2
1524          2053 2
1525          2054 2
1526          2055 2
1527          2056 2
1528          2057 2
1529          2058 2
1530          2059 2
1531          2060 2
1532          2061 2
1533          2062 2
1534          2063 2
1535          2064 2
1536          2065 2
1537          2066 2
1538          2067 2
1539          2068 2
1540          2069 2
1541          2070 2
1542          2071 2
1543          2072 2
1544          2073 2
1545          2074 2
1546          2075 2
1547          2076 2
1548          2077 2
1549          2078 2
1550          2079 2
1551          2080 2
1552          2081 2
1553          2082 2
1554          2083 2
1555          2084 2
1556          2085 2
1557          2086 2
1558          2087 2
1559          2088 2
1560          2089 2
1561          2090 2
1562          2091 2
1563          2092 2
1564          2093 2
1565          2094 2
1566          2095 2
1567          2096 2
1568          2097 2
1569          2098 2
1570          2099 2
1571          2100 2
1572          2101 2
1573          2102 2
1574          2103 2
1575          2104 2
1576          2105 2
1577          2106 2
1578          2107 2
1579          2108 2
1580          2109 2
1581          2110 2
1582          2111 2
1583          2112 2
1584          2113 2
1585          2114 2
1586          2115 2
1587          2116 2
1588          2117 2
1589          2118 2
1590          2119 2
1591          2120 2
1592          2121 2
1593          2122 2
1594          2123 2
1595          2124 2
1596          2125 2
1597          2126 2
1598          2127 2
1599          2128 2
1600          2129 2
1601          2130 2
1602          2131 2
1603          2132 2
1604          2133 2
1605          2134 2
1606          2135 2
1607          2136 2
1608          2137 2
1609          2138 2
1610          2139 2
1611          2140 2
1612          2141 2
1613          2142 2
1614          2143 2
1615          2144 2
1616          2145 2
1617          2146 2
1618          2147 2
1619          2148 2
1620          2149 2
1621          2150 2
1622          2151 2
1623          2152 2
1624          2153 2
1625          2154 2
1626          2155 2
1627          2156 2
1628          2157 2
1629          2158 2
1630          2159 2
1631          2160 2
1632          2161 2
1633          2162 2
1634          2163 2
1635          2164 2
1636          2165 2
1637          2166 2
1638          2167 2
1639          2168 2
1640          2169 2
1641          2170 2
1642          2171 2
1643          2172 2
1644          2173 2
1645          2174 2
1646          2175 2
1647          2176 2
1648          2177 2
1649          2178 2
1650          2179 2
1651          2180 2
1652          2181 2
1653          2182 2
1654          2183 2
1655          2184 2
1656          2185 2
1657          2186 2
1658          2187 2
1659          2188 2
1660          2189 2
1661          2190 2
1662          2191 2
1663          2192 2
1664          2193 2
1665          2194 2
1666          2195 2
1667          2196 2
1668          2197 2
1669          2198 2
1670          2199 2
1671          2200 2
1672          2201 2
1673          2202 2
1674          2203 2
1675          2204 2
1676          2205 2
1677          2206 2
1678          2207 2
1679          2208 2
1680          2209 2
1681          2210 2
1682          2211 2
1683          2212 2
1684          2213 2
1685          2214 2
1686          2215 2
1687          2216 2
1688          2217 2
1689          2218 2
1690          2219 2
1691          2220 2
1692          2221 2
1693          2222 2
1694          2223 2
1695          2224 2
1696          2225 2
1697          2226 2
1698          2227 2
1699          2228 2
1700          2229 2
1701          2230 2
1702          2231 2
1703          2232 2
1704          2233 2
1705          2234 2
1706          2235 2
1707          2236 2
1708          2237 2
1709          2238 2
1710          2239 2
1711          2240 2
1712          2241 2
1713          2242 2
1714          2243 2
1715          2244 2
1716          2245 2
1717          2246 2
1718          2247 2
1719          2248 2
1720          2249 2
1721          2250 2
1722          2251 2
1723          2252 2
1724          2253 2
1725          2254 2
1726          2255 2
1727          2256 2
1728          2257 2
1729          2258 2
1730          2259 2
1731          2260 2
1732          2261 2
1733          2262 2
1734          2263 2
1735          2264 2
1736          2265 2
1737          2266 2
1738          2267 2
1739          2268 2
1740          2269 2
1741          2270 2
1742          2271 2
1743          2272 2
1744          2273 2
1745          2274 2
1746          2275 2
1747          2276 2
1748          2277 2
1749          2278 2
1750          2279 2
1751          2280 
```

```

: 1263      1792 3      THEN
: 1264      1793 4      BEGIN
: 1265      1794 4      extend_outfile = TRUE;           ! set a flag saying that the file is being extend
: 1266      1795 4
: 1267      1796 4      output_xaball [xab$1_alq] =
: 1268      1797 4          copy$calc_alq ();           ! Calculate the necessary extension quantity
: 1269      1798 4
: 1270      1799 4      IF .output_xaball [xab$1_alq] NEQ 0   ! If the extension quantity is not null,
: 1271      1800 4      THEN
: 1272      P 1801 4          IF NOT SRMS_EXTEND (           ! then try to extend the file.
: 1273      P 1802 4              FAB = .output_fab,
: 1274      1803 5              ERR = copy$oufopen_err)
: 1275      1804 4      THEN
: 1276      1805 4          RETURN no_file;           ! If the extend fails,
: 1277      1806 4
: 1278      1807 4      END
: 1279      1808 4
: 1280      1809 3      ELSE
: 1281      1810 4      BEGIN
: 1282      1811 4          copy$log_msg (
: 1283      1812 4              msg$_overlay);           ! If this is a COPY command,
: 1284      1813 4
: 1285      1814 4      ****
: 1286      1815 4          Omitted here is the revision of the output file's attributes. Ward had this
: 1287      1816 4          commented out.
: 1288      1817 4      ****
: 1289      1818 4
: 1290      1819 3      END;
: 1291      1820 3
: 1292      1821 2      END;
: 1293      1822 2
: 1294      1823 2      TES;                                ! End of SELECT expression.
: 1295      1824 2
: 1296      1825 2
: 1297      1826 2      Return to the caller with a success status code.
: 1298      1827 2
: 1299      1828 2
: 1300      1829 2      RETURN ok;                            ! Return with a success code.
: 1301      1830 1      END;

```

		.PSECT SPLIT\$,NOWRT,NOEXE,2	
	00 00 2A 3B 00020 P.AAE:	.ASCII \;*!<0><0>	:
		.EXTRN SY\$CREATE, SY\$DISPLAY	
		.EXTRN SY\$EXTEND	
		.PSECT SCODE\$,NOWRT,2	
	OFFC 00000	.ENTRY COPY\$OPN OUTFIL, Save R2,R3,R4,R5,R6,R7,R8,-: 1393	
SB	0000G	CF 9E 00002	R9,R10,RT1
SA	0000G	CF 9E 00007	COPY\$CLI_STATUS, R11
SE	08	C2 0000C	COPY\$SEM_STATUS, R10
SS	04	AC DD 0000F	SUBL2 #8 SP
		MOVL OUTPUT_FAB, R3	

56	28	A3	DO	00013	MOVL	40(R3), R6	1473	
50	24	A3	DO	00017	MOVL	36(R3), R0	1475	
55	04	A0	DO	0001B	MOVL	4(R0), R5	1477	
58	04	A5	DO	0001F	MOVL	4(R5), R8	1479	
59	04	A8	DO	00023	MOVL	4(R8), R9	1491	
02	A1	01	E1	00027	BBC	#1, COPY\$SEM_STATUS+2, 1\$	1494	
0000V	CF	08	AC	DD	PUSHL	OUTPUT_RAB		
		01	FB	0002F	CALLS	#1, SETUP_EXTEND		
		04	00034		RET			
1D	S2	0C	AC	DO	00035	18:	1493	
36	A3	1D	A2	B0	00039	MOVL	INPUT_FAB, R2	1500
38	A3	36	A2	B0	0003E	MOVW	29(R2), 29(R3)	1502
1F	A3	38	A2	DO	00043	MOVW	54(R2), 54(R3)	1503
3E	A3	1F	A2	90	00048	MOVL	56(R2), 56(R3)	1504
48	A3	3E	A2	B0	0004D	MOVB	31(R2), 31(R3)	1506
04	A2	48	A2	B0	00052	MOVW	62(R2), 62(R3)	1507
04	A3	FF7FFDFF	8F	CB	00057	BICL3	#-8389121, 4(R2), R0	1514
			50	C8	00060	BISL2	R0, 4(R3)	
		1D	A2	95	00064	TSTB	29(R2)	
16	A3	40	8F	88	00069	BNEQ	2\$	1522
16	A3	40	09	11	0006E	BISB2	#64, 22(R3)	1524
16	A3	20	88	00070	BRB	3\$		
40	A3	40	8F	8A	00074	BISB2	#32, 22(R3)	1527
3C	A3	05	E1	00079	BICB2	#64, 22(R3)	1528	
		3C	A2	B0	0007E	BBC	#5, 64(R2), 4\$	1535
		03	03	11	00083	MOVW	60(R2), 60(R3)	1537
57	34	A3	B4	00085	BRB	5\$		
04	01	A6	9E	00088	CLRW	60(R3)	1539	
6A	40	A7	E9	0008C	MOVAB	52(R6), R7	1546	
67	40	8F	88	00090	BLBC	1(R7), 6\$	1548	
6A	03	F1	00094	6\$:	BISB2	#64, COPY\$SEM_STATUS	1551	
6A	80	8F	88	00098	BBC	#3, (R7), 7\$	1553	
		06	11	0009C	BISB2	#128, COPY\$SEM_STATUS		
03	67	E9	0009E	7\$:	BRB	8\$		
6A	04	88	000A1	8\$:	BLBC	(R7), 8\$	1555	
2C	68	E8	000A4	8\$:	BISB2	#4, COPY\$SEM_STATUS	1557	
6A	03	E0	000A7		BLBS	COPY\$CLI_STATUS, 13\$	1569	
6A	05	E0	000AB		BBS	#3, COPY\$SEM_STATUS, 12\$	1571	
6A	01	E1	000AF		BBC	#5, COPY\$SEM_STATUS, 9\$	1573	
67	03	E0	000B3	9\$:	BBS	#1, COPY\$SEM_STATUS, 10\$		
0F	67	E9	000B7		BLBC	#3, (R7), 10\$	1574	
16	67	E8	000BA	10\$:	BLBS	(R7), 12\$	1575	
67	04	E0	000BD		BBS	(R7), 13\$	1577	
67	01	E0	000C1		BBS	#4, (R7), 11\$	1578	
30	A3	0000	05	E1	000C5	BBC	#1, (R7), 13\$	1579
35	A3	CF	9E	000C9	11\$:	MOVAB	#5, (R7), 13\$	1582
		02	90	000CF	12\$:	MOVB	P_AAE, 48(R3)	1583
		0000G	CF	9F	000D3	13\$:	#2, 5\$(R3)	1589
			53	DD	000D7	PUSHAB	COPY\$OUTOPN_ERR	
			02	FB	000D9	PUSHL	R3	
00000000G	00	50	E9	000E0	CALLS	#2, SYSSPARSE		
37	50	A6	9A	000E3	BLBC	R0, 15\$		
50	48	A6	C0	000E7	MOVZBL	58(R6), R0	1599	
2A	FE	A0	91	000EB	ADDL2	72(R6), R0	1600	
2E	FE	A0	06	13	000EF	CMPB	-2(R0), #42	
			13	000F1	BEQL	14\$		
			91	000F1	CMPB	-2(R0), #46	1601	

H 16  
15-Sep-1984 23:42:51 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:14:19 [COPY.SRC]COPYSPEC\$ B32;1

15-Sep-1984 23:42:51 VAX-11 Bliss-32 v4.0-762  
14-Sep-1984 12:14:19 [COPY.SRC][COPYSPEC\$..B32;1]

{ 1  
15-Sep-1984 23:42:51 VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 12:14:19 [COPY.SRC]COPY\$SPEC\$.\$B32;1

Page 36  
(6)

02	22		6B	E9	002B1		BLBC	COPYSCLI_STATUS, 328	1791
0000G	AA	80	8F	88	002B4		BISB2	#128, COPYSEM_STATUS+2	1794
10	CF		00	FB	002B9		CALLS	#0, COPYSCALC_ALQ	1797
	AS		50	D0	002BE		MOVL	R0, 16(R5)	
			1C	13	002C2		BEQL	348	1799
		0000G	CF	9F	002C4		PUSHAB	COPYSOUTOPN_ERR	1803
			53	DD	002C8		PUSHL	R3	
00000000G	00		02	FB	002CA		CALLS	#1, SYSSEXEND	
	0C		50	E8	002D1		BLRS	R0, 348	
			0E	11	002D4		BRB	358	1805
0000G	7E	10AB	8F	3C	002D6	328:	MOVZWL	#4267, -(SP)	1811
	CF		01	FB	002DB	338:	CALLS	#1, COPYSLG_MSG	1829
	50		01	D0	002E0	348:	MOVL	#1, R0	
			04	002E3			RET		1830
			50	D4	002E4	358:	CLRL	R0	
				04	002E6		RET		

; Routine Size: 743 bytes, Routine Base: SCODES + 025C

```

1831 1 ROUTINE setup_extend (output_rab) = ! Setup a file to be extended.
1832 1
1833 1    ++
1834 1    Functional description:
1835 1
1836 1        This routine takes an open file and prepares it to be extended.
1837 1
1838 1        First, a DISCONNECT is performed. This permits switching from block mode I/O
1839 1        to record mode I/O, if desired. Then update the output file allocation information,
1840 1        set a bit in COPY$CLI_STATUS saying that the file is being extended, calculate
1841 1        the file extension quantity, and extend the file.
1842 1
1843 1    Calling sequence:
1844 1
1845 1        setup_extend (output_rab.ra.v)
1846 1
1847 1    Input parameters
1848 1
1849 1        output_rab      - the RAB connected to the output FAB
1850 1
1851 1    Implicit inputs
1852 1
1853 1        The FAB and XAB blocks associated with the specified output RAB block.
1854 1
1855 1    Output parameters
1856 1
1857 1        none
1858 1
1859 1    Implicit outputs
1860 1
1861 1        The allocation information in the FAB is updated.
1862 1        The EXTEND_DUTFILE bit in COPY$CLI_STATUS is set.
1863 1        The ALQ field in the output XAB block is set to an appropriate extension quantity.
1864 1
1865 1    Routine value
1866 1
1867 1        OK          - success
1868 1        NO_FILE     - failure
1869 1
1870 1    Side effects
1871 1
1872 1        If the file cannot be extended, the file is closed.
1873 1
1874 1    --
1875 1
1876 2    BEGIN
1877 2
1878 2    MAP
1879 2        output_rab      : REF BLOCK [, BYTE];           ! output FAB of the open output file
1880 2
1881 2    BIND
1882 2        output_fab      =
1883 2            .output_rab [rab$1_fab]                 : BLOCK [, BYTE],
1884 2            .output_xabfhc [fab$1_xab]                 : BLOCK [, BYTE],
1885 2            .output_xaball [xab$1_nxt]                : BLOCK [, BYTE],
1886 2            .output_xabfhe [xab$1_nxt]               : BLOCK [, BYTE];
1887 2

```

```
: 1360      1888      2
: 1361      1889      2
: 1362      1890      2
: 1363      1891      2
: 1364      1892      2
: 1365      1893      2
: 1366      1894      2
: 1367      1895      2
: 1368      1896      2
: 1369      1897      2
: 1370      1898      2
: 1371      1899      2
: 1372      1900      2
: 1373      1901      2
: 1374      1902      P
: 1375      1903      P
: 1376      1904      P
: 1377      1905      P
: 1378      1906      P
: 1379      1907      P
: 1380      1908      P
: 1381      1909      P
: 1382      1910      P
: 1383      1911      P
: 1384      1912      P
: 1385      1913      P
: 1386      1914      P
: 1387      1915      P
: 1388      1916      P
: 1389      1917      P
: 1390      1918      P
: 1391      1919      P
: 1392      1920      P
: 1393      1921      P
: 1394      1922      P
: 1395      1923      P
: 1396      1924      P
: 1397      1925      P
: 1398      1926      P
: 1399      1927      P
: 1400      1928      P
: 1401      1929      P
: 1402      1930      P
: 1403      1931      P
: 1404      1932      P
: 1405      1933      P
: 1406      1934      P
: 1407      1935      P
: 1408      1936      P
: 1409      1937      P
: 1410      1938      P
: 1411      1939      P
: 1412      1940      P
: 1413      1941      P
: 1414      1942      P
: 1415      1943      P
: 1416      1944      P

    LOCAL status;                                ! Holds RMS status values

    ! See if the input file fits the criteria given on the command line.

    IF NOT (status = copy$check_file_for_match())
    THEN
        RETURN .status;

    ! Disconnect the RAB from the FAB. On error, close the file and return
    ! with error status code.

    IF NOT SRMS_DISCONNECT (
        RAB = .output_rab,
        ERR = copy$oclose_err)
    THEN
        BEGIN
            copy$close_outf (
                output_fab);
        RETURN no_file;
        END;

    ! Disconnect the output file RAB from its FAB.
    ! Specify the RAB block address
    ! and an error routine.

    ! If the DISCONNECT fails,
    ! close the output file,
    ! and return with an error code.

    ! Shortening the XAB chain to include only the FHC (file header characteristics) XAB,
    ! call the RMS function SDISPLAY to update the output file allocation information
    ! as recorded in the XABFHC.

    output_xabfhc [xab$1_nxt] = 0;                ! Leave only the FHC XAB on the XAB chain.

    status = SRMS_DISPLAY (
        FAB = output_fab,
        ERR = copy$outopn_err);                    ! Call DISPLAY to update the XAB information
                                                    ! about the file's allocation.
                                                    ! Specify an error action routine.

    output_xabfhc [xab$1_nxt] = output_xaball;     ! Restore the XAB chain.

    ! See if the SDISPLAY function succeeded. If not, close the output file and return
    ! an error status code.

    IF NOT .status
    THEN
        BEGIN
            copy$close_outf (
                output_fab);
        RETURN no_file;
        END;

    ! If the SDISPLAY function failed,
    ! then close the output file,
    ! and return an error status code.

    ! Set the bit in COPY$CLI_STATUS that indicates that the file is to be extended.

    extend_outfile = TRUE;                         ! Set EXTEND_OUTFILE bit.
```

```

1417 1945 2
1418 1946 2
1419 1947 2
1420 1948 2
1421 1949 2
1422 1950 2
1423 1951 2
1424 1952 2
1425 1953 2
1426 1954 2
1427 1955 2
1428 1956 2
1429 1957 2
1430 1958 2
1431 1959 2
1432 P 1960 2
1433 P 1961 2
1434 1962 3
1435 1963 2
1436 1964 2
1437 1965 2
1438 1966 2
1439 1967 2
1440 1968 1

      Calculate the file extension quantity and extend the file with an RMS SEXTEND function call.  

      The routine COPYSCALC_ALQ does the calculation. It returns a "zero" in the following cases:  

          The output file is on a magtape or a nonfile-structured device.  

          The output file is already long enough to hold the size of the file to be appended.

      output_xaball [xab$1_alq] = copy$calc_alq ();           ! Setup the output file extension quantity in the XA
      IF .output_xaball [xab$1_alq] EQ 0                      ! If the input file is of zero length,
      THEN                                                 ! then return with success code.
          RETURN ok;
      IF SRMS_EXTEND (                                     ! If the output file can be extended successfully,
          FAB = output_fab,                                ! (specify an error routine)
          ERR = copy$outrpn_err)
      THEN
          RETURN ok
      ELSE
          RETURN no_file;
      END;

```

.EXTRN SYSSDISCONNECT

007C 00000 SETUP_EXTEND:									
54	04	AC	DO	00002	.WORD	MOVL	Save R2,R3,R4,R5,R6		1831
55	3C	A4	DO	00006		MOVL	OUTPUT_RAB, R4		1883
52	24	A5	DO	0000A		MOVL	60(R4), R5		1885
53	04	A2	DO	0000E		MOVL	36(R5), R2		1887
0000G	CF	00	FB	00012		CALLS	#0. COPY\$CHECK_FILE_FOR_MATCH		1895
56	50	DO	00017			MOVL	R0. STATUS		
04	56	E8	0001A			BLBS	STATUS, 1\$		
50	56	DO	0001D			MOVL	STATUS, R0		1897
			04	00020		RET			
	0000G	CF	9F	00021	1\$:	PUSHAB	COPY\$OCLOSE_ERR		1905
00000000G	00	54	DD	00025		PUSHL	R4		
1A	02	FB	00027			CALLS	#2. SYSSDISCONNECT		
	50	E9	0002E			BLBC	R0, 2\$		
	04	A2	D4	00031		CLRL	4(R2)		1919
	0000G	CF	9F	00034		PUSHAB	COPY\$OUTOPN_ERR		1923
00000000G	00	55	DD	00038		PUSHL	R5		
56	02	FB	0003A			CALLS	#2. SYSSDISPLAY		
04	A2	50	DO	00041		MOVL	R0. STATUS		
09	53	DO	00044			MOVL	R3, 4(R2)		1925
	56	E8	00048			BLBS	STATUS, 3\$		1932
	0000G	CF	55	DD	0004B	2\$:	PUSHL	R5	1935
		01	FB	0004D		CALLS	#1. COPY\$CLOSE_OUTF		
0000G	CF	25	11	00052		BRB	5\$		1937
0000G	CF	80	8F	00054	3\$:	BISB2	#128. COPY\$SEM_STATUS+2		1944
10	00	FB	0005A			CALLS	#0. COPY\$CALC_ALQ		1954
	A3	50	DO	0005F		MOVL	R0, 16(R3)		

	0000G	10	13 00063	BEQL	6S	
		CF	9F 00065	PUSHAB	COPY\$OUTOPN_ERR	: 1956
		55	DD 00069	PUSHL	R5	: 1962
00000000G	00	02	FB 00068	CALLS	#2, SYS\$EXTEND	
	04	50	E9 00072	BLBC	R0, SS	
	50	01	D0 00075	48:	MOVL #1, R0	1966
			04	00078	RET	
			50	D4 00079	58:	CLRL R0
			04	00078		1968

; Routine Size: 124 bytes.    Routine Base: SCODE\$ + 0543

```
: 1442      1 ROUTINE setup_outxab (output_fab, input_fab) : NOVALUE =
: 1443      1                                         ! Setup output XAB fields from input XAB fields
: 1444
: 1445
: 1446
: 1447
: 1448
: 1449
: 1450
: 1451
: 1452
: 1453
: 1454
: 1455
: 1456      1 ** Functional description:
: 1457      1
: 1458      1   This routine copies input XAB fields into corresponding output XAB fields.
: 1459
: 1460
: 1461      1 Calling sequence:
: 1462      1
: 1463      1   setup_outxab (output_fab.ra.v, input_fab.ra.v)
: 1464
: 1465      1 Input parameters:
: 1466
: 1467      1   output_fab      - FAB block associated with the output file
: 1468      1   input_fab       - FAB block associated with the input file
: 1469
: 1470      1 Implicit inputs:
: 1471
: 1472      1   output_xaball    - XABALL block for output file
: 1473      1   output_xabdat   - XABDAT block for output file
: 1474      1   output_xabfhc   - XABFHC block for output file
: 1475      1   output_xabpro   - XABPRO block for output file
: 1476      1   output_xabrdt   - XABRDT block for output file
: 1477
: 1478      1   input_xaball    - XABALL block for input file
: 1479      1   input_xabdat   - XABDAT block for input file
: 1480      1   input_xabfhc   - XABFHC block for input file
: 1481      1   input_xabpro   - XABPRO block for input file
: 1482
: 1483
: 1484
: 1485
: 1486
: 1487
: 1488
: 1489
: 1490
: 1491
: 1492
: 1493      1 Output parameters
: 1494      1
: 1495      1   none
: 1496
: 1497
: 1498
: 1499
: 1500
: 1501
: 1502
: 1503
: 1504
: 1505
: 1506
: 1507
: 1508
: 1509
: 1510
: 1511
: 1512
: 1513
: 1514
: 1515
: 1516
: 1517
: 1518
: 1519
: 1520
: 1521
: 1522
: 1523
: 1524
: 1525
: 1526
: 1527
: 1528
: 1529
: 1530
: 1531
: 1532
: 1533
: 1534
: 1535
: 1536
: 1537
: 1538
: 1539
: 1540
: 1541
: 1542
: 1543
: 1544
: 1545
: 1546
: 1547
: 1548
: 1549
: 1550
: 1551
: 1552
: 1553
: 1554
: 1555
: 1556
: 1557
: 1558
: 1559
: 1560
: 1561
: 1562
: 1563
: 1564
: 1565
: 1566
: 1567
: 1568
: 1569
: 1570
: 1571
: 1572
: 1573
: 1574
: 1575
: 1576
: 1577
: 1578
: 1579
: 1580
: 1581
: 1582
: 1583
: 1584
: 1585
: 1586
: 1587
: 1588
: 1589
: 1590
: 1591
: 1592
: 1593
: 1594
: 1595
: 1596
: 1597
: 1598
: 1599
: 1600
: 1601
: 1602
: 1603
: 1604
: 1605
: 1606
: 1607
: 1608
: 1609
: 1610
: 1611
: 1612
: 1613
: 1614
: 1615
: 1616
: 1617
: 1618
: 1619
: 1620
: 1621
: 1622
: 1623
: 1624
: 1625
: 1626
: 1627
: 1628
: 1629
: 1630
: 1631
: 1632
: 1633
: 1634
: 1635
: 1636
: 1637
: 1638
: 1639
: 1640
: 1641
: 1642
: 1643
: 1644
: 1645
: 1646
: 1647
: 1648
: 1649
: 1650
: 1651
: 1652
: 1653
: 1654
: 1655
: 1656
: 1657
: 1658
: 1659
: 1660
: 1661
: 1662
: 1663
: 1664
: 1665
: 1666
: 1667
: 1668
: 1669
: 1670
: 1671
: 1672
: 1673
: 1674
: 1675
: 1676
: 1677
: 1678
: 1679
: 1680
: 1681
: 1682
: 1683
: 1684
: 1685
: 1686
: 1687
: 1688
: 1689
: 1690
: 1691
: 1692
: 1693
: 1694
: 1695
: 1696
: 1697
: 1698
: 1699
: 1700
: 1701
: 1702
: 1703
: 1704
: 1705
: 1706
: 1707
: 1708
: 1709
: 1710
: 1711
: 1712
: 1713
: 1714
: 1715
: 1716
: 1717
: 1718
: 1719
: 1720
: 1721
: 1722
: 1723
: 1724
: 1725
: 1726
: 1727
: 1728
: 1729
: 1730
: 1731
: 1732
: 1733
: 1734
: 1735
: 1736
: 1737
: 1738
: 1739
: 1740
: 1741
: 1742
: 1743
: 1744
: 1745
: 1746
: 1747
: 1748
: 1749
: 1750
: 1751
: 1752
: 1753
: 1754
: 1755
: 1756
: 1757
: 1758
: 1759
: 1760
: 1761
: 1762
: 1763
: 1764
: 1765
: 1766
: 1767
: 1768
: 1769
: 1770
: 1771
: 1772
: 1773
: 1774
: 1775
: 1776
: 1777
: 1778
: 1779
: 1780
: 1781
: 1782
: 1783
: 1784
: 1785
: 1786
: 1787
: 1788
: 1789
: 1790
: 1791
: 1792
: 1793
: 1794
: 1795
: 1796
: 1797
: 1798
: 1799
: 1800
: 1801
: 1802
: 1803
: 1804
: 1805
: 1806
: 1807
: 1808
: 1809
: 1810
: 1811
: 1812
: 1813
: 1814
: 1815
: 1816
: 1817
: 1818
: 1819
: 1820
: 1821
: 1822
: 1823
: 1824
: 1825
: 1826
: 1827
: 1828
: 1829
: 1830
: 1831
: 1832
: 1833
: 1834
: 1835
: 1836
: 1837
: 1838
: 1839
: 1840
: 1841
: 1842
: 1843
: 1844
: 1845
: 1846
: 1847
: 1848
: 1849
: 1850
: 1851
: 1852
: 1853
: 1854
: 1855
: 1856
: 1857
: 1858
: 1859
: 1860
: 1861
: 1862
: 1863
: 1864
: 1865
: 1866
: 1867
: 1868
: 1869
: 1870
: 1871
: 1872
: 1873
: 1874
: 1875
: 1876
: 1877
: 1878
: 1879
: 1880
: 1881
: 1882
: 1883
: 1884
: 1885
: 1886
: 1887
: 1888
: 1889
: 1890
: 1891
: 1892
: 1893
: 1894
: 1895
: 1896
: 1897
: 1898
: 1899
: 1900
: 1901
: 1902
: 1903
: 1904
: 1905
: 1906
: 1907
: 1908
: 1909
: 1910
: 1911
: 1912
: 1913
: 1914
: 1915
: 1916
: 1917
: 1918
: 1919
: 1920
: 1921
: 1922
: 1923
: 1924
: 1925
: 1926
: 1927
: 1928
: 1929
: 1930
: 1931
: 1932
: 1933
: 1934
: 1935
: 1936
: 1937
: 1938
: 1939
: 1940
: 1941
: 1942
: 1943
: 1944
: 1945
: 1946
: 1947
: 1948
: 1949
: 1950
: 1951
: 1952
: 1953
: 1954
: 1955
: 1956
: 1957
: 1958
: 1959
: 1960
: 1961
: 1962
: 1963
: 1964
: 1965
: 1966
: 1967
: 1968
: 1969
: 1970
: 1971
: 1972
: 1973
: 1974
: 1975
: 1976
: 1977
: 1978
: 1979
: 1980
: 1981
: 1982
: 1983
: 1984
: 1985
: 1986
: 1987
: 1988
: 1989
: 1990
: 1991
: 1992
: 1993
: 1994
: 1995
: 1996
: 1997
: 1998
: 1999
: 2000
: 2001
: 2002
: 2003
: 2004
: 2005
: 2006
: 2007
: 2008
: 2009
: 2010
: 2011
: 2012
: 2013
: 2014
: 2015
: 2016
: 2017
: 2018
: 2019
: 2020
: 2021
: 2022
: 2023
: 2024
: 2025
: 2026
: 2027
: 2028
: 2029
: 2030
: 2031
: 2032
: 2033
: 2034
: 2035
: 2036
: 2037
: 2038
: 2039
: 2040
: 2041
: 2042
: 2043
: 2044
: 2045
: 2046
: 2047
: 2048
: 2049
: 2050
: 2051
: 2052
: 2053
: 2054
: 2055
: 2056
: 2057
: 2058
: 2059
: 2060
: 2061
: 2062
: 2063
: 2064
: 2065
: 2066
: 2067
: 2068
: 2069
: 2070
: 2071
: 2072
: 2073
: 2074
: 2075
: 2076
: 2077
: 2078
: 2079
: 2080
: 2081
: 2082
: 2083
: 2084
: 2085
: 2086
: 2087
: 2088
: 2089
: 2090
: 2091
: 2092
: 2093
: 2094
: 2095
: 2096
: 2097
: 2098
: 2099
: 2100
: 2101
: 2102
: 2103
: 2104
: 2105
: 2106
: 2107
: 2108
: 2109
: 2110
: 2111
: 2112
: 2113
: 2114
: 2115
: 2116
: 2117
: 2118
: 2119
: 2120
: 2121
: 2122
: 2123
: 2124
: 2125
: 2126
: 2127
: 2128
: 2129
: 2130
: 2131
: 2132
: 2133
: 2134
: 2135
: 2136
: 2137
: 2138
: 2139
: 2140
: 2141
: 2142
: 2143
: 2144
: 2145
: 2146
: 2147
: 2148
: 2149
: 2150
: 2151
: 2152
: 2153
: 2154
: 2155
: 2156
: 2157
: 2158
: 2159
: 2160
: 2161
: 2162
: 2163
: 2164
: 2165
: 2166
: 2167
: 2168
: 2169
: 2170
: 2171
: 2172
: 2173
: 2174
: 2175
: 2176
: 2177
: 2178
: 2179
: 2180
: 2181
: 2182
: 2183
: 2184
: 2185
: 2186
: 2187
: 2188
: 2189
: 2190
: 2191
: 2192
: 2193
: 2194
: 2195
: 2196
: 2197
: 2198
: 2199
: 2200
: 2201
: 2202
: 2203
: 2204
: 2205
: 2206
: 2207
: 2208
: 2209
: 2210
: 2211
: 2212
: 2213
: 2214
: 2215
: 2216
: 2217
: 2218
: 2219
: 2220
: 2221
: 2222
: 2223
: 2224
: 2225
: 2226
: 2227
: 2228
: 2229
: 2230
: 2231
: 2232
: 2233
: 2234
: 2235
: 2236
: 2237
: 2238
: 2239
: 2240
: 2241
: 2242
: 2243
: 2244
: 2245
: 2246
: 2247
: 2248
: 2249
: 2250
: 2251
: 2252
: 2253
: 2254
: 2255
: 2256
: 2257
: 2258
: 2259
: 2260
: 2261
: 2262
: 2263
: 2264
: 2265
: 2266
: 2267
: 2268
: 2269
: 2270
: 2271
: 2272
: 2273
: 2274
: 2275
: 2276
: 2277
: 2278
: 2279
: 2280
: 2281
: 2282
: 2283
: 2284
: 2285
: 2286
: 2287
: 2288
: 2289
: 2290
: 2291
: 2292
: 2293
: 2294
: 2295
: 2296
: 2297
: 2298
: 2299
: 2300
: 2301
: 2302
: 2303
: 2304
: 2305
: 2306
: 2307
: 2308
: 2309
: 2310
: 2311
: 2312
: 2313
: 2314
: 2315
: 2316
: 2317
: 2318
: 2319
: 2320
: 2321
: 2322
: 2323
: 2324
: 2325
: 2326
: 2327
: 2328
: 2329
: 2330
: 2331
: 2332
: 2333
: 2334
: 2335
: 2336
: 2337
: 2338
: 2339
: 2340
: 2341
: 2342
: 2343
: 2344
: 2345
: 2346
: 2347
: 2348
: 2349
: 2350
: 2351
: 2352
: 2353
: 2354
: 2355
: 2356
: 2357
: 2358
: 2359
: 2360
: 2361
: 2362
: 2363
: 2364
: 2365
: 2366
: 2367
: 2368
: 2369
: 2370
: 2371
: 2372
: 2373
: 2374
: 2375
: 2376
: 2377
: 2378
: 2379
: 2380
: 2381
: 2382
: 2383
: 2384
: 2385
: 2386
: 2387
: 2388
: 2389
: 2390
: 2391
: 2392
: 2393
: 2394
: 2395
: 2396
: 2397
: 2398
: 2399
: 2400
: 2401
: 2402
: 2403
: 2404
: 2405
: 2406
: 2407
: 2408
: 2409
: 2410
: 2411
: 2412
: 2413
: 2414
: 2415
: 2416
: 2417
: 2418
: 2419
: 2420
: 2421
: 2422
: 2423
: 2424
: 2425
: 2426
: 2427
: 2428
: 2429
: 2430
: 2431
: 2432
: 2433
: 2434
: 2435
: 2436
: 2437
: 2438
: 2439
: 2440
: 2441
: 2442
: 2443
: 2444
: 2445
: 2446
: 2447
: 2448
: 2449
: 2450
: 2451
: 2452
: 2453
: 2454
: 2455
: 2456
: 2457
: 2458
: 2459
: 2460
: 2461
: 2462
: 2463
: 2464
: 2465
: 2466
: 2467
: 2468
: 2469
: 2470
: 2471
: 2472
: 2473
: 2474
: 2
```

```

: 1499      2026 2      output_xabfhc =          | output XAB file header characteristics block
: 1500      2027 2      .output_fab [fab$1_xab] : BLOCK [, BYTE];
: 1501      2028 2      output_xaball =          | output XAB date block
: 1502      2029 2      .output_xabfhc [xab$1_nxt] : BLOCK [, BYTE];
: 1503      2030 2      output_xabdat =          | output XAB date block
: 1504      2031 2      .output_xaball [xab$1_nxt] : BLOCK [, BYTE];
: 1505      2032 2      output_xabrdt =          | output XAB date block
: 1506      2033 2      .output_xabdat [xab$1_nxt] : BLOCK [, BYTE];
: 1507      2034 2      output_xabpro =          | output XAB date block
: 1508      2035 2      .output_xabrdt [xab$1_nxt] : BLOCK [, BYTE];
: 1509      2036 2
: 1510      2037 2      input_xaball =           | input file XABALL block
: 1511      2038 2      .input_fab [fab$1_xab] : BLOCK [, BYTE];
: 1512      2039 2      input_xabdat =          | input file XABDAT block
: 1513      2040 2      .input_xaball [xab$1_nxt] : BLOCK [, BYTE];
: 1514      2041 2      input_xabfhc =          | input file XABFHC block
: 1515      2042 2      .input_xabdat [xab$1_nxt] : BLOCK [, BYTE];
: 1516      2043 2      input_xabpro =          | input file XABPRO block
: 1517      2044 2      .input_xabfhc [xab$1_nxt] : BLOCK [, BYTE];
: 1518      2045 2
: 1519      2046 2      | Write the output allocation XAB.
: 1520      2047 2
: 1521      2048 2
: 1522      2049 2
: 1523      2050 2      output_xaball [xab$b_aop] =
: 1524      2051 2      .input_xaball [xab$b_aop];   ! Write the allocation options,
: 1525      2052 2      output_xaball [xab$b_aln] =
: 1526      2053 2      .input_xaball [xab$b_aln];   ! and the alignment type.
: 1527      2054 2
: 1528      2055 2      output_xaball [xab$1_alq] = copy$calc_alq (); ! Calculate and write in the allocation quantity.
: 1529      2056 2
: 1530      2057 2      output_xaball [xab$w_deq] =
: 1531      2058 2      .input_xabfhc [xab$w_dxq]; ! Write the default extension quantity.
: 1532      2059 2      output_xaball [xab$b_bkz] =
: 1533      2060 2      .input_fab [fab$b_bks];   ! Write the default bucket size
: 1534      2061 2
: 1535      2062 2
: 1536      2063 2
: 1537      2064 2
: 1538      2065 2      output_xaball [xab$w_vol] = 0; ! Zero the related volume number,
: 1539      2066 2      output_xaball [xab$1_loc] = 0; ! the allocation location,
: 1540      2067 2      output_xaball [xab$b_aid] = 0; ! the area id number,
: 1541      2068 2      output_xaball [xab$w_rf10] = 0; ! the related file number
: 1542      2069 2      output_xaball [xab$w_rf12] = 0; ! the related file sequence number
: 1543      2070 2      output_xaball [xab$w_rf14] = 0; ! and the related file revision number.
: 1544      2071 2
: 1545      2072 2      IF .input_fab [$fab_dev(net)] AND ! If this is a network operation
: 1546      2073 2      .output_xaball [xab$1_alq] EQL 0 and the calculated ALQ = 0,
: 1547      2074 2      THEN output_xaball [xab$1_alq] = then get ALQ from the FHC XAB
: 1548      2075 2      .input_xabfhc [xab$1_hbk];
: 1549      2076 2
: 1550      2077 2      | Write the output Date/Time XAB.
: 1551      2078 2
: 1552      2079 2
: 1553      2080 2
: 1554      2081 2      output_xabdat [xab$w_rvn] = ! Increment the revision number
: 1555      2082 2      .input_xabdat [xab$w_rvn] + 1;

```

```
: 1556      2083 2      output_xabdat [xab$1_rdt0] = 0;          ! Clear the revision date
: 1557      2084 2      output_xabdat [xab$1_rdt4] = 0;          ! Copy the creation date
: 1558      2085 2      output_xabdat [xab$1_cdt0] =
: 1559      2086 2      .input_xabdat [xab$1_cdt0];
: 1560      2087 2      output_xabdat [xab$1_cdt4] =
: 1561      2088 2      .input_xabdat [xab$1_cdt4];
: 1562      2089 2
: 1563      2090 2      These values are not copied from the input, but defaulted instead,
: 1564      2091 2      so the user will get new backup and expiration dates.
: 1565      2092 2
: 1566      2093 2
: 1567      2094 2      ! If the output device is tape, then propagate the expiration date.
: 1568      2095 2      ! Otherwise, clear it.
: 1569      2096 2
: 1570      2097 2      IF .output_fab[ $FAB_DEV(sqd) ]
: 1571      2098 2      THEN
: 1572      2099 2      BEGIN
: 1573      2100 2      output_xabdat [xab$1_edt0] = .input_xabdat [xab$1_edt0];
: 1574      2101 2      output_xabdat [xab$1_edt4] = .input_xabdat [xab$1_edt4];
: 1575      2102 2      END
: 1576      2103 2      ELSE
: 1577      2104 2      BEGIN
: 1578      2105 2      output_xabdat [xab$1_edt0] = 0;
: 1579      2106 2      output_xabdat [xab$1_edt4] = 0;
: 1580      2107 2      END;
: 1581      2108 2
: 1582      2109 2      output_xabdat [xab$1_bdt0] = 0;          ! the backup date
: 1583      2110 2      output_xabdat [xab$1_bdt4] = 0;          ! and the backup time
: 1584      2111 2
: 1585      2112 2
: 1586      2113 2      Write the output File Header Characteristics XAB block.
: 1587      2114 2
: 1588      2115 2
: 1589      2116 2      output_xabfhc [xab$2_rfo] =
: 1590      2117 2      .input_xabfhc [xab$2_rfo];
: 1591      2118 2      output_xabfhc [xab$2_atr] =
: 1592      2119 2      .input_xabfhc [xab$2_atr];
: 1593      2120 2      output_xabfhc [xab$2_lrl] =
: 1594      2121 2      .input_xabfhc [xab$2_lrl];
: 1595      2122 2      output_xabfhc [xab$2_bkz] =
: 1596      2123 2      .input_xabfhc [xab$2_bkz];
: 1597      2124 2      output_xabfhc [xab$2_hsz] =
: 1598      2125 2      .input_xabfhc [xab$2_hsz];
: 1599      2126 2      output_xabfhc [xab$2_mrz] =
: 1600      2127 2      .input_xabfhc [xab$2_mrz];
: 1601      2128 2      output_xabfhc [xab$2_dxq] =
: 1602      2129 2      .input_xabfhc [xab$2_dxq];
: 1603      2130 2
: 1604      2131 2      output_xabfhc [xab$1_sbn] = 0;          ! Zero the starting virtual block number.
: 1605      2132 2
: 1606      2133 2
: 1607      2134 2      Write the output Protection XAB block. Most of this XAB can only be setup
: 1608      2135 2      after the output file has been opened or created. Therefore, it is not done here.
: 1609      2136 2
: 1610      2137 2
: 1611      2138 2      output_xabpro [xab$1_uic] = 0;          ! Clear the file owner field.
: 1612      2139 2
```

```
1613 2140 2 |  
1614 2141 2 | Write the output Revision Date/Time XAB block.  
1615 2142 2 |  
1616 2143 2 |  
1617 2144 2 | output_xabrdt [xab$w_rvn] = ! Increment revision number  
1618 2145 2 | .input_xabdat [xab$w_rvn ] + 1;  
1619 2146 2 | output_xabrdt [xab$1_rdt0] = 0; ! Do not propogate the the input revision date,  
1620 2147 2 | output_xabrdt [xab$1_rdt4] = 0;  
1621 2148 2 |  
1622 2149 2 | *****  
1623 2150 2 | Temporarily, I omit the special saving of XABDAT and XABFHC fields  
1624 2151 2 | of a file that may be overwritten. This must go back in.  
1625 2152 2 | *****  
1626 2153 2 |  
1627 2154 1 | END:
```

1A A6	1A A5 B0 0008B	MOVW 26(R5), 26(R6)
	28 A6 D4 00090	CLRL 40(R6)
D8 A9	0C AA D4 00093	CLRL 12(R10)
	0C 50 B0 00096	MOVW R0, 8(R9)
	0C A9 7C 0009A	CLRQ 12(R9)
	04 0009D	RET

: 2129  
: 2131  
: 2138  
: 2145  
: 2146  
: 2154

; Routine Size: 158 bytes, Routine Base: SCODES + 05BF

1629 2155 1 ROUTINE apply\_out\_qual (output\_fab) : NOVALUE = ! Applies output parameter qualifiers to FAB and XAB  
1630 2156 1  
1631 2157 1  
1632 2158 1 ++ Functional description  
1633 2159 1  
1634 2160 1 This routine looks for the presence of qualifiers on the output file specification,  
1635 2161 1 and sets RMS fields according to the semantics of each qualifier.  
1636 2162 1  
1637 2163 1 Calling sequence:  
1638 2164 1      apply\_out\_qual (output\_fab.ra.v)  
1639 2165 1  
1640 2166 1 Input parameters:  
1641 2167 1      output\_fab - the FAB block related to the output file specification  
1642 2168 1  
1643 2169 1 Implicit inputs:  
1644 2170 1      output\_xaball - The XABALL block associated with the output FAB  
1645 2171 1  
1646 2172 1  
1647 2173 1 The following bits in COPY\$CLI\_STATUS:  
1648 2174 1  
1649 2175 1  
1650 2176 1  
1651 2177 1  
1652 2178 1  
1653 2179 1  
1654 2180 1  
1655 2181 1  
1656 2182 1  
1657 2183 1  
1658 2184 1  
1659 2185 1  
1660 2186 1  
1661 2187 1  
1662 2188 1  
1663 2189 1 Some values associated with qualifiers specified for the output file specification:  
1664 2190 1  
1665 2191 1  
1666 2192 1  
1667 2193 1  
1668 2194 1  
1669 2195 1  
1670 2196 1  
1671 2197 1  
1672 2198 1  
1673 2199 1 Output parameters  
1674 2200 1  
1675 2201 1  
1676 2202 1  
1677 2203 1  
1678 2204 1  
1679 2205 1 Some fields in the output XABALL block are written:  
1680 2206 1  
1681 2207 1  
1682 2208 1  
1683 2209 1  
1684 2210 1  
1685 2211 1  
ALN - alignment type  
AOP - alignment option  
LOC - alignment location  
ALQ - allocation quantity  
CTG - contiguous file

1686 2212 1 |  
1687 2213 1 |  
1688 2214 1 |  
1689 2215 1 |  
1690 2216 1 |  
1691 2217 1 |  
1692 2218 1 |  
1693 2219 1 |  
1694 2220 1 |  
1695 2221 1 |  
1696 2222 1 |  
1697 2223 1 |  
1698 2224 1 |  
1699 2225 1 |  
1700 2226 1 |  
1701 2227 1 |  
1702 2228 1 |  
1703 2229 1 |  
1704 2230 1 |  
1705 2231 1 |  
1706 2232 1 |  
1707 2233 1 |  
1708 2234 1 |  
1709 2235 2 |  
1710 2236 2 |  
1711 2237 2 |  
1712 2238 2 |  
1713 2239 2 |  
1714 2240 2 |  
1715 2241 2 |  
1716 2242 2 |  
1717 2243 2 |  
1718 2244 2 |  
1719 2245 2 |  
1720 2246 2 |  
1721 2247 2 |  
1722 2248 2 |  
1723 2249 2 |  
1724 2250 2 |  
1725 2251 2 |  
1726 2252 2 |  
1727 2253 2 |  
1728 2254 2 |  
1729 2255 2 |  
1730 2256 2 |  
1731 2257 2 |  
1732 2258 2 |  
1733 2259 2 |  
1734 2260 2 |  
1735 2261 2 |  
1736 2262 2 |  
1737 2263 2 |  
1738 2264 2 |  
1739 2265 2 |  
1740 2266 2 |  
1741 2267 2 |  
1742 2268 3 |

CBT = contiguous best try file  
DEQ = file extension quantity  
VOL = relative volume number

Some fields in the output FAB are written:

MRN = maximum record number  
CIF = create if nonexistent file  
RCK = read check  
TEF = truncate files at EOF mark  
SUP = supersede  
WCK = write check

Routine value  
novalue

Side effects  
none

--

BEGIN

MAP

    output\_fab : REF BLOCK [, BYTE]; ! Output file FAB block

BIND

    output\_nam = .output\_fab [fab\$1\_nam] : BLOCK [, BYTE]; ! output NAM block address  
    output\_xabfhc = .output\_fab [fab\$1\_xab] : BLOCK [, BYTE]; ! output XAB file header characteristics block  
    output\_xaball = .output\_xabfhc [xab\$1\_nxt] : BLOCK [, BYTE]; ! output XAB date block  
    output\_xabdat = .output\_xaball [xab\$1\_nxt] : BLOCK [, BYTE]; ! output XAB date block  
    output\_xabrdt = .output\_xabdat [xab\$1\_nxt] : BLOCK [, BYTE]; ! output XAB date block  
    output\_xabpro = .output\_xabrdt [xab\$1\_nxt] : BLOCK [, BYTE]; ! output XAB date block

Apply the effects of the output file qualifiers to the appropriate XAB blocks.

: /ALLOCATION = n  
: IF qualifier\_active( alloc\_qual, loc\_alloc\_qual, neg\_alloc\_qual )  
THEN  
    output\_xaball [xab\$1\_sq] = .curr\_allocation\_value;  
IF qualifier\_active( contig\_qual, loc\_contig\_qual, neg\_contig\_qual )  
THEN  
    BEGIN  
        output\_xaball [xab\$1\_ctg] = TRUE;  
        output\_xaball [xab\$1\_cbt] = FALSE;

```

1743 2269 1
1744 2270 2
1745 2271 3
1746 2272 4
1747 2273 5
1748 2274 6
1749 2275 7
1750 2276 8
1751 2277 9
1752 2278 2
1753 2279 2
1754 2280 2
1755 2281 2
1756 2282 2
1757 2283 2
1758 2284 2
1759 2285 2
1760 2286 2
1761 2287 2
1762 2288 2
1763 2289 2
1764 2290 2
1765 2291 2
1766 2292 2
1767 2293 2
1768 2294 2
1769 2295 2
1770 2296 2
1771 2297 3
1772 2298 2
1773 2299 2
1774 2300 2
1775 2301 2
1776 2302 2
1777 2303 2
1778 2304 2
1779 2305 2
1780 2306 2
1781 2307 2
1782 2308 2
1783 2309 2
1784 2310 2
1785 2311 2
1786 2312 2
1787 2313 2
1788 2314 2
1789 2315 2
1790 2316 2
1791 2317 2
1792 2318 2
1793 2319 2
1794 2320 2
1795 2321 2
1796 2322 2
1797 2323 1

    ELSE END
    BEGIN
    IF .config_negated OR .neg_config_qual
    THEN
        BEGIN
        output_xaball [xab$v_ctg] = FALSE;
        output_xaball [xab$v_cbt] = FALSE;
        END;
    END;

    IF qualifier_active( extend_qual, loc_extend_qual, neg_extend_qual )
    THEN
        output_xaball [xab$w_deq] = .curr_extension_value;

    IF qualifier_active( file_max_qual, loc_file_max_qual, neg_file_max_qual )
    THEN
        output_fab [fab$1_mrn] = .curr_file_max_value;

    IF qualifier_active( overlay_qual, loc_overlay_qual, neg_overlay_qual ) OR
        .new_version_qual
    THEN
        output_fab [fab$v_cif] = TRUE;

    IF qualifier_active( replace_qual, loc_replace_qual, neg_replace_qual )
    THEN
        output_fab [fab$v_sup] = TRUE;

    IF qualifier_active( truncate_qual, loc_truncate_qual, neg_truncate_qual )
    THEN
        output_fab [fab$v_tef] = TRUE;

    IF qualifier_active( volume_qual, loc_volume_qual, neg_volume_qual )
    THEN
        BEGIN
        output_xaball [xab$w_vol] = .curr_volume_value;
        output_xaball [xab$b_sln] = xab$c_lbn;
        output_xaball [xab$v_hrd] = 1;
        END;

    IF qualifier_active( write_chk_qual, loc_write_chk_qual, neg_write_chk_qual )
    THEN
        output_fab [fab$v_wck] = TRUE
    ELSE
        BEGIN
        IF .write_chk_negated
        THEN
            output_fab [fab$v_wck] = FALSE;
        END;

    ! Return to caller.

    END;

```

0000 00000 APPLY_OUT_QUAL:									
53	0000G	CF	9E	00002	QORD	Save R2, R3			2155
51	04	AC	00	00007	MOVAB	COPY\$CLI_STATUS+4, R3			2242
50	24	A1	00	0000B	MOVL	OUTPUT_FAB, R1			2244
50	04	A0	00	0000F	MOVL	3E(R1), R0			2246
52	04	A9	00	00013	MOVL	4(R0), R0			2248
05	FE	A3	E9	00017	BLBC	4(R0), R2			2260
06	FE	A3	02	E1 0001B	BBC	COPY\$CLI_STATUS+2, 1S			
10	A0	01	E1	00020	1S:	#2, COPY\$CLI_STATUS+2, 2S			2262
05	FE	A3	CF	00	00025	2S:	MOVL	CURR ALLOCATION VALUE, 16(R0)	2264
05	FE	A3	03	E1	0002B	3S:	BBC	#1, COPY\$CLI_STATUS+2, 3S	
07	FE	A3	06	E1	00030	BBC	#3, COPY\$CLI_STATUS+2, 4S		
08	A0	05	E1	00035	4S:	BBC	#6, COPY\$CLI_STATUS+2, 5S		
08	A0	80	8F	88 0003A	5S:	BISB2	#5, COPY\$CLI_STATUS+2, 6S		
08	A0	0F	11	0003F	BRB	#128, 8(R0)			2267
09	FE	A3	04	E0	00041	6S:	BBS	#4, COPY\$CLI_STATUS+2, 7S	2268
09	FE	A3	06	E1	00046	7S:	BBC	#6, COPY\$CLI_STATUS+2, 9S	2272
08	A0	80	8F	8A 0004B	7S:	BICB2	#128, 8(R0)		2275
08	A0	20	8A	00050	8S:	BICB2	#32, 8(R0)		2276
		FE	A3	95	00054	9S:	TSTB	COPY\$CLI_STATUS+2	2280
04	FF	A3	05	18	00057	BGEQ	10S		
	06	FF	A3	01	E1 00059	BBC	#1, COPY\$CLI_STATUS+3, 11S		
14	A0	0000G	CF	80	00062	10S:	BLBC	COPY\$CLI_STATUS+3, 12S	2282
05	FF	A3	02	E1	00068	11S:	MOVJ	CURR EXTENSION VALUE, 20(R0)	2284
05	FF	A3	04	E1	0006D	12S:	BBC	#2, COPY\$CLI_STATUS+3, 13S	
06	FF	A3	03	E1	00072	13S:	BBC	#4, COPY\$CLI_STATUS+3, 14S	
38	A1	0000G	CF	00	00077	14S:	MOVL	#3, COPY\$CLI_STATUS+3, 15S	2286
			63	95	0007D	15S:	TSTB	CURR FILE_MAX VALUE, 56(R1)	2288
			05	18	0007F	BGEQ	COPY\$CLI_STATUS+4		
09	01	A3	01	E1	00081	BBC	16S	#1, COPY\$CLI_STATUS+5, 17S	
04	FC	A3	05	A3	E8 00086	16S:	BLBS	COPY\$CLI_STATUS+5, 17S	
07	A1	04	E1	0008A	BBC	#4, COPY\$CLI_STATUS, 18S			2289
02	A3	02	88	0008F	17S:	BISB2	#2, 7(R1)		2291
05	02	A3	01	E1	00093	18S:	BBC	#1, COPY\$CLI_STATUS+6, 19S	2293
05	02	A3	03	E1	00098	BBC	#3, COPY\$CLI_STATUS+6, 20S		
04	02	A3	02	E1	0009D	19S:	BBC	#2, COPY\$CLI_STATUS+6, 21S	
04	04	A1	04	88	000A2	20S:	BISB2	#4, 4(R1)	2295
04	01	A3	05	E1	000A6	21S:	BBC	#5, COPY\$CLI_STATUS+5, 22S	2297
	05	05	02	A3	E9 000AB	22S:	BLBC	COPY\$CLI_STATUS+6, 23S	
		01	A3	95	000AF	22S:	TSTB	COPY\$CLI_STATUS+5	
			04	18	000B2	BGEQ	24S		
05	07	A1	10	88	000B4	23S:	BISB2	#16, 7(R1)	2299
05	01	A3	02	E1	000B8	24S:	BBC	#2, COPY\$CLI_STATUS+5, 25S	2301
05	01	A3	04	E1	000BD	BBC	#4, COPY\$CLI_STATUS+5, 26S		
0E	01	A3	03	E1	000C2	25S:	BBC	#3, COPY\$CLI_STATUS+5, 27S	
0A	A0	0000G	CF	80	000C7	26S:	MOVW	CURR_VOLUME_VALUE, 10(R0)	2304
09	A0	02	90	000CD	MOVW	#2, 9(R0)			2305
08	A0	01	88	000D1	BISB2	#1, 8(R0)			2306
04	63	03	E1	000D5	27S:	BBC	#3, COPY\$CLI_STATUS+4, 28S	2309	
04	63	06	E1	000D9	27S:	BBC	#6, COPY\$CLI_STATUS+4, 29S		
05	63	05	E1	000DD	28S:	BBC	#5, COPY\$CLI_STATUS+4, 30S		
	05	A1	02	88	000E1	29S:	BISB2	#2, 5(R1)	2311

04	05	63	04	000E5	RET	
			02	E1 000E6 30\$:	BBC	#4, COPY\$CLI_STATUS+4, 31\$
			04	BA 000EA	BICB2	#2, 5(R1)
			04	000EE 31\$:	RET	

; 2314  
; 2316  
; 2323

; Routine Size: 239 bytes,   Routine Base: \$CODE\$ + 0650

: 1799              2324 1 END  
: 1800              2325 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLIT\$	36 NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	
\$CODE\$	1868 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_S255\$DUA28:[SYSLIB]STARLET.L32;1	9776	179	1	581	00:01.0

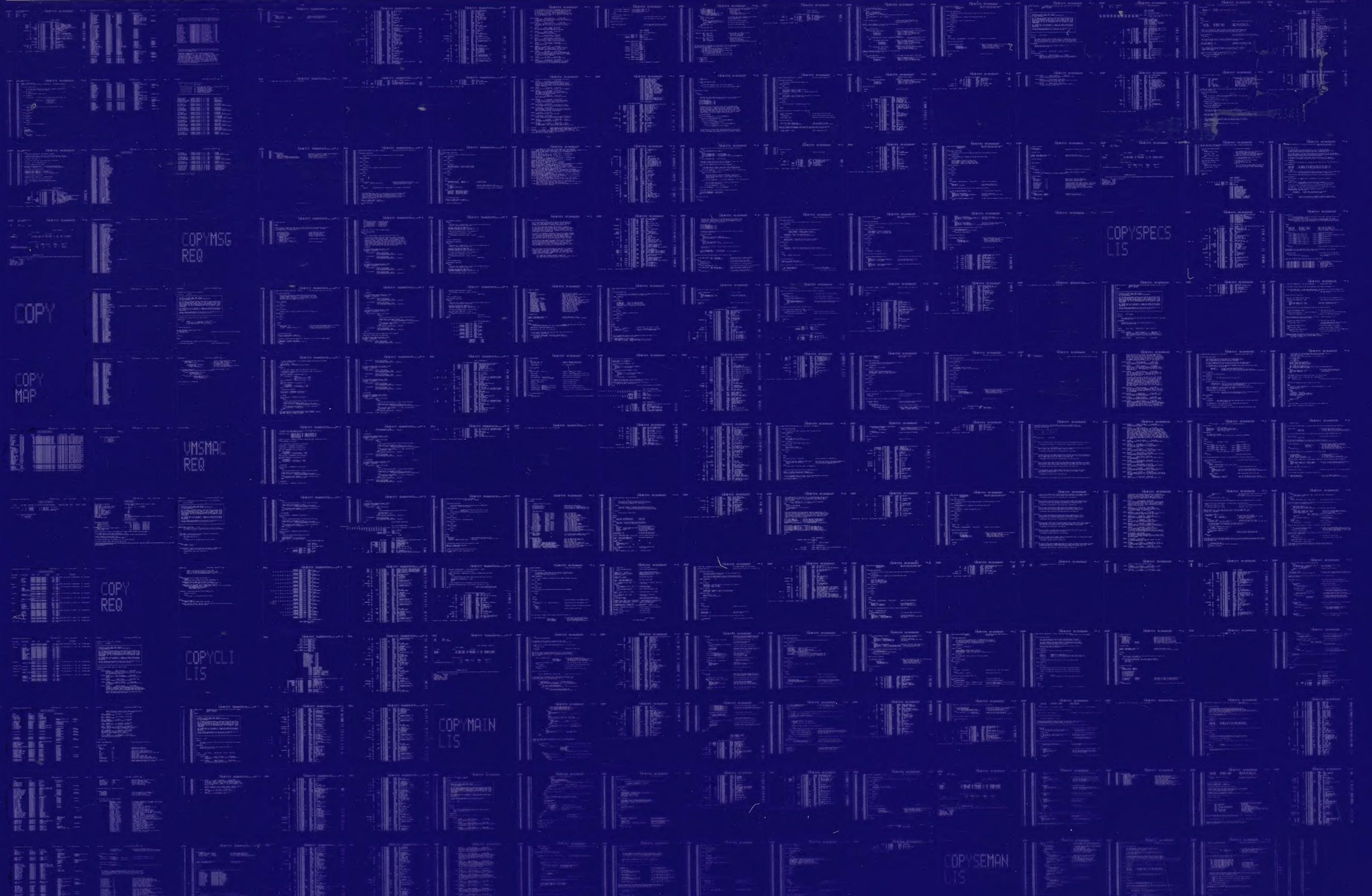
COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:COPYSPEC\$/OBJ=OBJ\$:COPYSPEC\$ MSRC\$:COPYSPEC\$/UPDATE=(ENH\$:COPYSPEC\$)

: Size:        1868 code + 36 data bytes  
: Run Time:    00:51.0  
: Elapsed Time: 02:08.1  
: Lines/CPU Min: 2735  
: Lexemes/CPU-Min: 28632  
: Memory Used: 286 pages  
: Compilation Complete

0067 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY



0068 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY